



**MEDIA PENGENALAN HURUF HIJAIYAH UNTUK TAMAN KANAK -
KANAK DENGAN *OUTPUT* SUARA DAN TANDA BACA
BERBASIS MIKROKONTROLER ATMEGA16**

LAPORAN PROYEK AKHIR

Diajukan Kepada Fakultas Teknik Universitas Negeri Yogyakarta

Untuk Memenuhi Sebagian Persyaratan Guna Memperoleh Gelar Ahli Madya



Disusun Oleh :

MULYANA

NIM. 10507131013

JURUSAN PENDIDIKAN TEKNIK ELEKTRONIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

2014

LEMBAR PERSETUJUAN


Proyek Akhir yang berjudul “Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16” ini telah disetujui oleh pembimbing untuk diujikan.

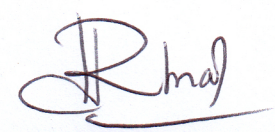


Yogyakarta, 17 September 2014

Mengetahui
Koord. Prodi D3 Teknik Elektronika,

Menyetujui
Dosen Pembimbing,


Drs. Djoko Santoso, M.Pd.
NIP. 19580422 198403 1 002


Dr. Ratna Wardani, S.Si, M.T
NIP. 19701218 200501 2 001

LEMBAR PERNGESAHAN

Tugas Proyek Akhir dengan Judul :

MEDIA PENGENALAN HURUF HIJAIYAH UNTUK TAMAN KANAK - KANAK DENGAN *OUTPUT* SUARA DAN TANDA BACA BERBASIS MIKROKONTROLER ATMEGA16

Disusun Oleh:
MULYANA
NIM. 10507131013

Telah dipertahankan di depan Tim Penguji Proyek Akhir Program Studi
Teknik Elektronika D3 Fakultas Teknik Universitas Negeri Yogyakarta

Pada Tanggal: 29 September 2014

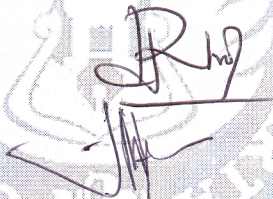
DEWAN PENGUJI

Nama/Jabatan

Tanda Tangan

Tanggal

Dr. Ratna Wardani, S.Si, M.T.
Ketua Penguji / Pembimbing



3/7 2015

Drs. Totok Sukardiyono, M.T.
Sekertaris



7/10 2014

Drs. Dr. Putu Sudira, M.P.
Penguji Utama



7/10 2014

Yogyakarta, September 2014
Fakultas Teknik Universitas Negeri Yogyakarta
Dekan,



Dr. Moch. Bruri Triyono, M. Pd.
NIP. 19560216 198603 1 003

SURAT PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini :

Nama : Mulyana

NIM : 10507131013

Program Studi : Teknik Elektronika (D3)

Judul Proyek Akhir : Media Pengenalan Huruf Hijaiyah untuk Taman Kanak -
Kanak dengan *Output* Suara dan Tanda Baca Berbasis
Mikrokontroler ATmega16.

Menyatakan bahwa Proyek Akhir ini adalah hasil pekerjaan saya sendiri dan sepanjang pengetahuan saya, tidak berisi materi yang ditulis orang lain sebagai persyaratan penyelesaian studi di Universitas Negeri Yogyakarta atau perguruan tinggi lain, kecuali bagian – bagian tertentu yang saya ambil sebagai acuan dengan mengikuti tata cara dan penulisan karya ilmiah yang lazim. Jika ternyata terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab saya.

Yogyakarta, 29 September 2014

Penulis,

Mulyana

(NIM. 10507131013)

MOTTO

Bismillahirrahmanirrahim

إِنَّ السَّمْعَ وَالْبَصَرَ وَالْفُؤَادَ كُلُّ أُولَئِكَ كَانَ عَنْهُ مَسْئُولًا

"Sesungguhnya pendengaran, penglihatan dan hati, semuanya itu akan dimintai pertanggung jawaban." (QS. Al-Isra' Ayat 36)

"Wahai orang – orang yang beriman! Jika kamu menolong (agama) Allah, niscaya Dia akan menolongmu dan meneguhkan kedudukanmu" (Qs. Muhammad Ayat 7)

"Dan bahwa manusia hanya memperoleh apa yang telah diusahakannya dan sesungguhnya usahanya itu kelak akan diperlihatkan (kepadanya), kemudian akan diberi balasan kepadanya dengan balasan yang paling sempurna" (QS. An-Najm Ayat 39-41)

PERSEMBAHAN

Dengan sepenuh hati penulis ingin mempersembahkan karya ini untuk :

Allah SWT tuhan seluruh alam dan pencipta alam semesta ini.

Dosen pembimbing Proyek Akhir, Dr. Ratna Wardani, S.Si., M.T yang telah sabar membimbing penulis dalam penyelesaian proyek akhir.

Ibu, bapak, kakak, adik dan teman – teman serta semua saudara yang telah memberikan do'a dan semangat sehingga penulis dapat menyelesaikan tugas akhir ini.

Seluruh teman – teman mahasiswa teknik elektronika-D3 angkatan 2010 yang telah memberikan support dan dukungannya.

Dukungan kakak tingkatan mahasiswa teknik elektronika yang sudah memberikan masukan dan kritiknya.

Semua ustad – ustadzah TKA/TPA Masjid Al – Falaah Mrican dan Santri – santri TKA/TPA Masjid Al – Falaah.

**Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan
Output Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16.**

MULYANA
NIM : 1050713013

ABSTRAK

Proyek akhir ini dimaksudkan untuk mengembangkan suatu perangkat berupa model pembelajaran yang mampu mengenalkan huruf - huruf hijaiyah beserta tanda bacanya yang ditampilkan dalam papan LCD dan *output* suara yang ditampilkan melalui *speaker*. Dari tampilan LCD akan mengeluarkan karakter huruf yang ditekan dan *speaker* akan mengeluarkan suara huruf.

Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16 ini, terdiri dari unit sistem minimum yang berfungsi sebagai pengendali rangkaian, unit *keyboard* sebagai perintah masukan, unit *MP3 player* sebagai penampil *file* suara, *MMC/SD Card* sebagai penyimpan data *file* suara, *power supply* sebagai pemberi tegangan keseluruhan rangkaian, unit LCD 16x2 sebagai keluaran penampil tanda baca huruf hijaiyah, unit *speaker* sebagai keluaran suara huruf hijaiyah, dari unit keseluruhan akan diproses dengan menggunakan sistem minimum mikrokontroler ATmega16. Perangkat ini digunakan untuk mengendalikan seluruh sistem. Perangkat lunak sebagai pengendali program pada mikrokontroler ATmega16 menggunakan *software* CVAVR sebagai tempat untuk membuat program dan *men-download* program ke dalam ATmega16.

Pada Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16 ini, menggunakan 30 tombol huruf hijaiyah dan 6 tombol sebagai tanda bacanya. Untuk menggunakan media pembelajaran huruf hijaiyah ini, hal pertama yang dilakukan ialah memilih tombol tanda baca huruf yang akan digunakan, lalu menekan tombol huruf hijaiyah. Keluaran pada LCD akan menampilkan karakter huruf hijaiyah secara bersamaan dengan suara dari *speaker*. Alat ini membutuhkan tegangan sebesar 12volt yang digunakan untuk *men-supply* tegangan ke rangkaian penguat *Amplifier* dan *MMC/SD Card*, *MP3 Player*. Tegangan akan dikecilkan dengan menggunakan IC regulator serta tegangan 5volt yang digunakan untuk menyuplai tegangan ke sistem minimum.

Kata Kunci : LCD, *MP3 Player*, ATmega16

KATA PENGANTAR

Alhamdulillah, puji syukur selalu dipanjatkan kehadiran Allah SWT atas rahmat dan hidayah-Nya, sehingga pada saat ini dapat terselesaikannya laporan proyek akhir ini dengan judul **“Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16”** Pembuatan Proyek Akhir ini tidak lepas dari bantuan berbagai pihak, baik itu secara langsung maupun tidak langsung.

Ucapan terima kasih kepada semua pihak yang telah memberikan bimbingan dan dukungan, sehingga pada saat ini Laporan Proyek Akhir telah selesai dibuat tepat waktu. Oleh karena itu pada kesempatan ini ucapan terima kasih yang sebesar-besarnya kepada :

1. Allah SWT yang atas rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir.
2. Bapak Prof. Dr. Rochmat Wahab, M. Pd, MA selaku rektor Universitas Negeri Yogyakarta.
3. Bapak Dr. Moch. Bruri Triyono, M. Pd. Selaku Dekan Fakultas Teknik UNY.
4. Bapak Muhammad Munir, M. Pd selaku Ketua Jurusan Pendidikan Teknik Elektronika Fakultas Teknik UNY.
5. Ibu Dr. Ratna Wardani, S.Si, M.T. selaku pembimbing Proyek Akhir.
6. Bapak Totok Sukardiyono, M.T selaku Sekertaris Penguji Proyek Akhir.
7. Bapak Putu Sudira, M.P. selaku Penguji Utama Proyek Akhir.
8. Bapak Djoko Santoso, M. Pd selaku Pembimbing Akademik.

9. Bapak, Ibu, Kakak dan Adik selaku pembimbing hidup.
10. Ustad ustadzah TKA/TPA Masjid Al - Falaah Mrican.
11. Santri – santri TKA/TPA Majid Al – Falaah Mrican.
12. Seluruh jamaah pengajian bapak – bapak dan ibu – ibu Masjid Al – Falaah Mrican.
13. Teman-teman mahasiswa angkatan 2010 Teknik Elektronika (D3).
14. Semua pihak yang telah membantu penulisan proyek akhir ini.

Dalam pembuatan Laporan Proyek Akhir ini perlu disadari bahwa setiap tulisan pada laporan ini pasti ada kekurangan dan jauh dari sempurna. Mengingat keterbatasan kemampuan dan pengetahuan yang di miliki, ungkapan kritik dan saran yang diharapkan dari pembaca sehingga laporan ini dapat lebih dengan baik lagi.

Yogyakarta, 29 September 2014

Penulis

Mulyana

(NIM. 10507131013)

DAFTAR ISI

	Halaman
HAMALAN JUDUL.....	i
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
LEMBAR PERNYATAAN KEASLIAN	iv
MOTTO.....	v
PERSEMBAHAN	vi
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN.....	xvi
BAB I PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Identifikasi Masalah	3
C. Batasan Masalah	4
D. Rumusan Masalah.....	4
E. Tujuan.....	5
F. Manfaat.....	5
G. Keaslian Gagasan	6
BAB II PENDEKATAN PEMECAHAN MASALAH	8
A. Pengertian Media	8
B. Pendidikan Taman Kanak-Kanak	9
C. Huruf Hijaiyah	9
1. Pengertian Huruf Hijaiyah	9
2. Bentuk – bentuk Huruf Hijaiyah.....	10
3. <i>Harokat</i> (Tanda Baca Huruf)	12
C. Mikrokontroler ATmega16.....	14

1. Arsitektur Mikrokontroler ATmega16.....	14
2. Konfigurasi ATmega16	17
D. <i>CodeVision AVR version 2.05.0</i>	20
E. <i>Proteus ISIS dan ARES</i>	21
F. <i>Keyboard</i>	22
G. Modul <i>MP3 Player</i>	25
H. <i>Micro SD Card</i> (Micro-Secure Digital Card)	26
1. Sistem Penulisan <i>Micro SD</i>	26
2. Sistem Penyimpanan <i>Micro SD</i>	26
3. Mengakses <i>File MP3</i> pada <i>SD Card</i> dengan Mikrokontroler	27
I. <i>LCD (Liquid Crystal Display)</i> 16x2	29
J. <i>IC Penguat Suara TDA7052</i>	32
K. <i>Speaker</i>	33
 BAB III KONSEP RANCANGAN	35
A. Identifikasi Kebutuhan.....	35
B. Analisis Kebutuhan.....	36
C. Perancangan Sistem	37
1. Perancangan Blok Diagram Rangkaian	37
2. Rangkaian Sistem Minimum ATmega16	37
3. Rangkaian Modul <i>MP3 Player</i> dan <i>Mini Power Amplifier</i>	38
4. Rangkaian Display LCD	39
5. Perancang Perangkat Lunak.....	40
a. Algoritma.....	40
b. Diagram Alir/ <i>Flowchart</i> Program.....	43
D. Pembuatan Alat.....	47
1. Alat dan Bahan.....	47
2. Pembuatan PCB	48
a. Pembuatan Layout PCB	48
b. Penyablonan PCB.....	48
c. Pelarutan dan Pengeboran PCB.....	49
3. Pemasangan Komponan.....	49

4. Pembuatan Rangkaian Box	50
a. Perancangan Ukuran Box	50
b. Pembuatan Rangkaian Box	50
E. Pengujian Alat.....	51
1. Pengujian Uji Fungsional Setiap Bagian	51
2. Pengujian Keseluruhan Uji Sistem	54
F. Jadwal Pelaksanaan.....	59
G. Daftar Komponen dan Biaya Pembuatan.....	60
 BAB IV HASIL, PENGUJIAN DAN PEMBAHASAN	62
A. Hasil	62
1. Hasil <i>Layout</i> PCB Rangkaian Sistem Minimum ATmega16, Rangkaian Penguat OP-AMP, Rangkaian Display LCD	62
2. Hasil Perancangan Alat.....	63
3. Hasil Perancangan Alat Jadi	63
B. Pengujian	64
1. Pengujian Stiap Bagian	64
2. Pengujian Keseluruhan Uji Sistem	72
C. Pembahasan	98
1. Pengukuran Tegangan Kerja	98
2. Pengujian Tombol Huruf Hijaiyah	99
3. Pengujian Kombinasi Huruf dan Tanda Baca huruf.....	100
4. Pengukuran Kecepatan Tombol <i>Input</i> dan <i>Output</i> Suara	101
D. Unjuk Kerja Alat.....	101
 BAB V KESIMPULAN DAN SARAN.....	103
A. Kesimpulan	103
B. Keterbatasan Alat.....	104
C. Saran	105
DAFTAR PUSTAKA	106
LAMPIRAN.....	108

DAFTAR TABEL

	Halaman
Tabel 1. Huruf Hijaiyah	11
Tabel 2. Tanda Baca Fathah.....	12
Tabel 3. Tanda Baca Fathatain.....	12
Tabel 4. Tanda Baca Kasroh	13
Tabel 5. Tanda Baca Kasrotain	13
Tabel 6. Tanda Baca Domah.....	13
Tabel 7. Tanda Baca Domatain.....	13
Tabel 8. Pin LCD dan Fungsinya.....	31
Tabel 9. Alat dan Bahan.....	47
Tabel 10. Pengujian Tegangan.....	52
Tabel 11. Pengujian Kode Tombol Keyboard.....	52
Tabel 12. Pengujian Tombol Huruf Hijaiyah.....	54
Tabel 13. Pengujian Tombol Huruf Hijaiyah Normal	55
Tabel 14. Pengujian Kombinasi Tombol Huruf Hijaiyah dan <i>Harokat</i>	57
Tabel 15. Pengukuran Kecepatan Tombol <i>Input</i> dan <i>Output</i> Suara.....	58
Tabel 16. Jadwal Pelaksanaan.....	59
Tabel 17. Daftar Komponen dan Biaya Pembuatan.....	60
Tabel 18. Data Pengukuran Tegangan	64
Tabel 19. Hasil Pengujian Tombol Kode Keyboard	66
Tabel 20. Hasil Pengujian Tombol Harokat.....	70
Tabel 21. Hasil Pengujian Tampilan Suara.....	72
Tabel 22. Hasil Pengujian Tombol Huruf Hijaiyah	72
Tabel 23. Hasil Pengujian Tombol Huruf Hijaiyah Normal	74
Tabel 24. Data Pengujian Kombinasi Huruf dan Tanda Baca Fathah	76
Tabel 25. Data Pengujian Kombinasi Huruf dan Tanda Baca Kasroh.....	78
Tabel 26. Data Pengujian Kombinasi Huruf dan Tanda Baca Domah	81
Tabel 27. Data Pengujian Kombinasi Huruf dan Tanda Baca Fathatain.....	83
Tabel 28. Data Pengujian Kombinasi Huruf dan Tanda Baca Kasrotain.....	85

Tabel 29. Data Pengujian Kombinasi Huruf dan Tanda Baca Domatain	87
Tabel 30. Data Pengukuran Kecepatan Tombol <i>Input</i> huruf hijaiyah dan <i>Ouput Suara</i>	89
Tabel 31. Data Pengukuran Kecepatan Tombol <i>Input Harokat</i> Fathah dan <i>Ouput Suara</i>	91
Tabel 32. Data Pengukuran Kecepatan Tombol <i>Input Harokat</i> Kasroh dan <i>Ouput Suara</i>	92
Tabel 33. Data Pengukuran Kecepatan Tombol <i>Input Harokat</i> Domah dan <i>Ouput Suara</i>	93
Tabel 34. Data Pengukuran Kecepatan Tombol <i>Input Harokat</i> Fathatain dan <i>Ouput Suara</i>	95
Tabel 35. Data Pengukuran Kecepatan Tombol <i>Input Harokat</i> Kasrotain dan <i>Ouput Suara</i>	94
Tabel 36. Data Pengukuran Kecepatan Tombol <i>Input Harokat</i> Domatain dan <i>Ouput Suara</i>	97

DAFTAR GAMBAR

	Halaman
Gambar 1. Blok Daigram ATmega16	14
Gambar 2. Konfigurasi Pin ATmega16	18
Gambar 3. <i>CodeVision AVR version 2.05.0</i>	21
Gambar 4. <i>Keyboard PS2 6 Connector</i>	23
Gambar 5. <i>Scan Code</i> Masing – masing Tombol <i>Keyboard PC</i>	24
Gambar 6. Konektor <i>Keyboard PS2</i> (MINI-DIN6)	24
Gambar 7. WT5001M02-28P Minimum.....	25
Gambar 8. Ilustrasi Pengaksesan <i>Low Level Disk I/O</i> Pada Modul <i>Fatfs</i>	28
Gambar 9. Koneksi Pin untuk antarmuka <i>SD Card/MMC</i> dengan Mikro-kontroler (Sistem SPI).....	28
Gambar 10. Blok Diagram LCD	29
Gambar 11. LCD Karakter 16x2	30
Gambar 12. IC TDA7052.....	32
Gambar 13. Penampang <i>Speaker</i>	33
Gambar 14. Blok Diagram Sistem	37
Gambar 15. Rangkaian Sistem Minimum.....	38
Gambar 16. Rangkaian Modul <i>MP3 Player</i> dan <i>Mini Power Amplifier</i>	39
Gambar 17. Rangkaian Display LCD 16x2	39
Gambar 18. <i>Flowchart</i>	46
Gambar 19. Desain Rangkaian Box	50
Gambar 20. Rangkaian Box	51
Gambar 21 Rangkaian Sistem Minimum ATmega16, Rangkaian Penguat OP-AMP, Rangkaian Display LCD	62
Gambar 22. Hasil Perancangan Alat	63
Gambar 23. Hasil Perancangan Alat Jadi.....	64
Gambar 24. Tampilan Awal LCD	69

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Skema Rangkaian Catu Daya	109
Lampiran 2. Skema Rangkaian <i>Amplifier</i>	110
Lampiran 3. Skema Rangkaian Keseluruhan (Sistem Minimum, Amplifier, Display LCD)	111
Lampiran 4. Program	112
Lampiran 5. Data Santri TKA/TPA Masjid Al-Falaah, Kelas Aisyah Belajar Mengenal Huruf Hijaiyah dengan Menggunakan Media Pembelajaran Huruf Hijaiyah	126
Lampiran 6. Atmel (ATmega16).....	131
Lampiran 7. Modul <i>MP3 Player</i> (WT5001M02-28P)	144
Lampiran 8. <i>MMC/SD Card</i>	145

BAB I

PENDAHULUAN

A. Latar Belakang

Islam menurut Mahmud Syalthout adalah Agama Allah yang diperintahkan kepada nabi Muhammad untuk mengajarkan pokok - pokok serta peraturan-peraturannya, serta menugaskan kepadanya untuk menyampaikan agama tersebut kepada seluruh manusia agar mereka memeluknya.

Setiap manusia yang memeluk Agama Islam wajib untuk belajar dan mengajarkan agama kepada sesamanya. Seperti halnya belajar membaca Al-Qur'an dan memahami isi kandungan kitab suci Al-Qur'an. Hal ini diwajibkan bagi seorang muslim yang sudah dewasa maupun masih anak - anak untuk belajar membaca Al-Qur'an. Di dalam belajar Al -Qur'an tidak mengenal batas umur dan usia, siapapun boleh mempelajarinya kitab Al-Qur'an, tidak ada larangan bagi setiap muslim lainnya untuk mempelajarinya. Namun sebelum mengenal bacaan Al-Qur'an, hendaknya terlebih dahulu mempelajari huruf hijaiyah yaitu tentang pengenalan *makhroj* dan sifat - sifat hurufnya. Huruf hijaiyah dapat dibaca walaupun tanpa menggunakan ke-6 tanda baca, sehingga bunyi hurufnya berupa suara aslinya. Dari ke-30 huruf hijaiyah ini ada beberapa huruf yang memiliki perbedaan bentuk dan sifat - sifatnya. Misalnya dari huruf *alif* dengan huruf *hamzah*, huruf *lam* dengan huruf *lam alif*, diantara kedua huruf ini memiliki bentuk huruf yang berbeda dan bila kedua huruf itu diberi tanda baca *fathah* maka bunyi hurufnya akan sama dan diakhir bacaan hurufnya ber-vocal "a". Begitupun dengan tanda baca yang lainnya seperti tanda baca *kasroh* yang jika digunakan

sebagai tanda baca untuk ke-30 huruf hijaiyah maka diakhir bacaan hurufnya bervokal “i”. Sedangkan tanda baca *domah* yang jika digunakan sebagai tanda baca huruf hijaiyah, maka di akhir bacaan hurufnya bervokal “u”.

Salah satu penerapan media pengenalan huruf hijaiyah ini merupakan suatu sarana untuk membantu proses belajar mengajar anak – anak dalam mengenal huruf hijaiyah. Dengan adanya media ini diharapkan dapat mempermudah proses pembelajaran anak untuk memahami materi yang diajarkan.

Dalam pembuatan media pembelajaran ini diperoleh hasil wawancara kepada seorang guru TKA/TPA Masjid Al-Falaah di daerah Mrican, Kecamatan Depok, Kabupaten Sleman, Yogyakarta. Diperoleh informasi bahwa kegiatan mengajar seringkali terjadi kendala pada anak - anak di dalam memahami materi yang disampaikan gurunya. Hal ini dimungkinkan karena faktor usia anak yang masih terlalu dini serta kejenuhan yang dialami oleh anak - anak membuat mereka malas untuk belajar. Anak - anak merasa bosan belajar karena metode belajar yang digunakan selalu menggunakan sistem pembelajaran tanya jawab yang dianggap terlalu *teoritis*. Pada waktu anak diajarkan oleh gurunya tentang mengenal huruf hijaiyah, hampir 60% anak tidak memahami bacaan huruf hijaiyah. Ada beberapa huruf yang memiliki kesamaan bunyi namun dari bentuk hurufnya berbeda seperti huruf “Sa” dengan huruf “Sya”, pengucapan huruf hampir sama tetapi dilihat dari *makhroj* terdapat perbedaan. Dalam membaca anak - anak juga terkadang bingung dengan huruf yang berbeda harokatnya hal ini membuat anak keliru dalam membacanya. Anak cenderung banyak bermain dengan teman – temannya dibandingkan dengan mendengarkan gurunya. Mereka berlari – lari dengan temannya, hal itu yang dikeluhkan oleh guru pengajar, tidak sedikit guru yang

mengundurkan diri karena mereka tidak betah mengajar. Beberapa sarana dan prasarana metode pembelajaran yang baru pun sudah dilakukan namun belum ada perubahan sampai saat ini.

Atas dasar itulah sebuah permasalahan dapat dihasilkan yaitu : permasalahan tentang membangun aplikasi “Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16”. Dengan adanya *aplikasi* ini diharapkan dapat membantu anak – anak agar lebih mudah serta menyenangkan dalam belajar mengenal huruf hijaiyah dengan baik dan benar. Media pengenalan huruf hijaiyah ini menggunakan tombol *keyboard* sebagai *input* perintah untuk memasukan huruf hijaiyah, sehingga jika salah satu tombol huruf ditekan maka *output* pada LCD 16 x 2 akan menampilkan tanda baca dari huruf bersamaan dengan bunyi huruf dari *Speaker*.

B. Identifikasi Masalah

Dari uraian latar belakang masalah ini, maka dapat dibuat suatu identifikasi masalah sebagai berikut :

1. Pembelajaran secara *teoritis* mengakibatkan anak merasa jenuh untuk belajar.
2. Hampir 60% anak - anak tidak bisa membaca huruf hijaiyah.
3. Kesamaan bunyi huruf membuat anak keliru membaca huruf hijaiyah.
4. Perbedaan harokat/tanda baca membuat anak kebingungan mengenali *mahroj* hurufnya.
5. Kurangnya prasarana media pembelajaran huruf hijaiyah berbasis mikrokontroler.

C. Batasan Masalah

Berdasarkan identifikasi masalah yang muncul, perlu adanya pembatasan masalah sehingga ruang lingkup permasalahannya menjadi jelas. Dalam pembuatan proyek akhir ini bermaksud untuk membuat sebuah alat yang berbasis mikrokontroler ATmega16 sebagai *control* utama seluruh rangkaian serta beberapa program pendukung seperti program pendeteksi *code* data pada tiap tombol *keyboard*, program untuk mendeteksi modul suara *MP3 player*, program untuk membaca *file* suara pada *micro-sd*, program penampil *display* karakter pada LCD, program penampil suara melalui *speaker*. Dari perpaduan antara *output* penampil *display* dan suara ini sehingga didapat suatu media pengenalan huruf hijaiyah yang sangat efektif dan efisien digunakan sebagai media pembelajaran pengenalan huruf hijaiyah bagi anak - anak.

D. Rumusan Masalah

Berdasarkan identifikasi dan batasan masalah yang telah disebutkan, maka permasalahan ini dapat dirumuskan sebagai berikut:

1. Bagaimana merancang *hardware* Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16?
2. Bagaimana merancang *software* pemrograman pada Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16?

3. Bagaimana unjuk kerja Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16?

E. Tujuan

Tujuan dari perancangan Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16 yaitu:

1. Merealisasikan *hardware* perancangan Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16 menggunakan Rangkaian Sistem Minimum ATmega16, Penguat OP-AMP, Catu Daya, Display LCD.
2. Merealisasikan *software* pada mikrokonroler ATmega16 dari Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16 menggunakan pemrograman *CV AVR Corporation*.
3. Merealisasikan sistem kerja dari perancangan Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16.

F. Manfaat

Penulis berharap dengan terciptanya Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16 dapat bermanfaat bagi:

1. Bagi Mahasiswa, merupakan suatu bentuk penerapan ilmu pengetahuan yang di dapat saat kuliah dalam suatu karya nyata.
2. Bagi Lembaga Instansi / Perguruan Tinggi, sebagai tolak ukur daya serap mahasiswa yang bersangkutan selama menempuh pendidikan dan kemampuan menerapkan ilmunya secara nyata, khususnya dalam mata kuliah mikrokontroler.
3. Bagi Masyarakat / Lembaga Pendidik, memberikan sumbangan berupa kontribusi bentuk karya teknologi penerapan bidang elektronika sebagai kebutuhan media pembelajaran bagi anak - anak.

G. Keaslian Gagasan

Pada tugas akhir yang berjudul “Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16” ini adalah benar – benar gagasan dan rancangan diri sendiri. Berdasarkan informasi yang diperoleh media pembelajaran huruf hijaiyah yang sejenisnya menggunakan *software macro media flash* dengan tanda baca *fathah, kasroh, domah, fathatain, kasrotain* dan *domatain*.

Perbedaan dari media pembelajaran yang pernah dibuat yaitu terletak dari penggunaan perangkatnya yang menggunakan perangkat *software macro media flash* dengan fitur animasi gambar dan rekaman suara huruf hijaiyah serta program pemanggil suara yang disimpan di dalam tiap-tiap *push button* huruf hijaiyah dan tanda baca. Pada perancangan media pembelajaran huruf hijaiyah yang dibuat pada saat ini menggunakan perangkat *hardware* dengan memanfaatkan mikrokontroler ATmega16 sebagai pengendalinya serta media

keyboard sebagai *input*-nya dan menggunakan perangkat display LCD serta suara dari *speaker* sebagai *output*-nya. Dengan terciptanya media pembelajaran huruf hijaiyah berbasis mikrokontroler ATmega16 ini diharapkan proses belajar anak-anak menjadi lebih *efektif* dan mudah dipahami dalam memahami dasar-dasar huruf hijaiyah.

BAB II

PENDEKATAN PEMECAHAN MASALAH

A. Pengertian Media

Perkembangan ilmu pengetahuan yang sangat pesat dapat berdampak pada pola hidup manusia yang semakin modern. Perkembangan ini membuat teknologi semakin berkembang, sehingga banyak alat-alat yang diciptakan untuk meringankan pekerjaan manusia. Misalnya dalam proses komunikasi dibutuhkan suatu media yang dapat digunakan untuk menyampaikan informasi kepada orang lain. Istilah media ini secara *harfiah* dapat diartikan sebagai perantara untuk menyampaikan pesan informasi kepada penerima. Gerlach & Ely (1971) berpendapat bahwa dalam membangun proses komunikasi dibutuhkan media untuk menyalurkan maksud dan tujuan berupa materi atau pengetahuan yang akan disampaikan oleh guru kepada siswa. Dari proses komunikasi tersebut media dapat dijadikan sebagai landasan umum agar tersampainya sebuah informasi. Suatu bentuk yang apabila memiliki tujuan untuk menyalurkan sebuah pesan dapat disebut sebagai media.

Proses belajar yang dilakukan dengan menggunakan media akan terjadinya perubahan tingkah laku pada diri seseorang di sebabkan karena faktor rasa keingintahuan yang tinggi yang mengakibatkan orang itu memiliki pengetahuan yang lebih serta keterampilan yang baik. Apabila sistem pembelajaran dilakukan di lingkungan sekolah dapat memungkinkan siswa lebih mengetahui tentang sumber ilmu pengetahuan yang sebenarnya. Karena di dalam lingkungan sekolah itulah yang menjadikan siswa menguasai ilmu pengetahuan.

B. Pendidikan Taman Kanak-Kanak

Menurut Ernawulan taman kanak-kanak adalah prose pendidikan bagi anak usia 4-6 tahun secara formal dapat ditempuh di taman kanak-kanak. Lembaga ini merupakan lembaga pendidikan yang ditunjukan untuk melaksanakan suatu proses pembelajaran agar anak dapat mengembangkan potensi-potensinya sejak dini hingga anak dapat berkembang secara wajar sebagaimana mestinya. Melalui suatu proses pembelajaran sejak usia dini, diharapkan anak tidak saja siap untuk memasuki jenjang pendidikan lebih lanjut, tetapi yang lebih utama agar anak memperoleh rangsangan-rangsangan fisik-motorik, intelektual, social dan emosi sesuai dengan tingkat usianya.

Anak usia dini adalah manusia individu yang sedang menjalani suatu proses perkembangan yang pesat dan fundamental bagi kehidupan selanjutnya. Anak memiliki dunia dan karakteristik tersendiri yang jauh berbeda dari dunia dan karakteristik orang dewasa. Anak sangat aktif, dinamis, antusias dan hampir selalu ingin tahu terhadap apa yang dilihat dan didengarnya. Melalui belajar anak akan berkembang dan mampu mempelajari hal - hal yang baru. Perkembangan anak dapat ditempuh dengan cara belajar, dari proses belajar ini anak akan memperoleh pengalaman dan pengetahuan baru.

C. Huruf Hijaiyah

1. Pengertian Huruf Hijaiyah

Menurut Sapiuddin (2012) huruf arab adalah huruf yang terdiri dari nama lambang, *makhraj* dan sifat-sifat huruf. *Makhraj* huruf yaitu tempat keluarnya suatu huruf yang diucapkan secara nyata, maka dengan adanya *makhraj* huruf ini

dapat dibedakan antara huruf satu dengan huruf yang lainnya. Sehingga huruf hijaiyah dapat dikatakan sebagai huruf ejaan bahasa arab sebagai bahasa asli Al-Qur'an.

Dalam proses pembelajaran huruf hijaiyah ini dapat dilakukan dengan mengenali materi dasar huruf hijaiyah. Materi dasar ini dapat dibagi menjadi dua jenis yaitu dengan mengenal *makhraj* dan sifat – sifat huruf hijaiyah. Kedua materi dasar ini adalah pokok utama untuk pengenalan huruf hijaiyah. Tanpa mengenal terlebih dahulu *makhraj* dan sifat – sifat huruf hijaiyah ini, kemungkinan besar orang itu akan keliru membacanya.

2. Bentuk - bentuk Huruf Hijaiyah

Huruf hijaiyah yang digunakan sebagai bahasa Al-Quran yang terdiri dari 30 macam huruf, jumlah tersebut termasuk *hamzah* dan *lam alif*. Pada dasarnya, huruf *alif* sama dengan *hamzah* dan huruf *lam alif* sama halnya dengan *lam*. Secara umum tempat keluarnya huruf hijaiyah berasal dari lima tempat yaitu :

- a. Rongga mulut (*Al - Jauf*) terdiri dari satu *makhraj*.
- b. Tenggorokan (*Al - Halq*) terdiri dari tiga *makhraj*.
- c. Lidah (*Al - Lisan*) terdiri dari sepuluh *makhraj*.
- d. Dua Bibir (*Asy - Syafatain*) terdiri dari dua *makhraj*.
- e. Hidung (*Al - Khaisyum*) terdiri dari satu *makhraj*.

Di bawah ini adalah bentuk – bentuk dari 30 huruf hijaiyah dengan cara membacanya yaitu sebagai berikut :

Tabel 1. Huruf Hijaiyah

No.	Huruf Hijaiyah	Dibaca
1.	ا	<i>Alif</i>
2.	ب	<i>Ba</i>
3.	ت	<i>Ta</i>
4.	ث	<i>Tsa</i>
5.	ج	<i>Jim</i>
6.	ح	<i>Kha</i>
7.	خ	<i>Kho</i>
8.	د	<i>Dal</i>
9.	ذ	<i>Dzal</i>
10.	ر	<i>Ro</i>
11.	ز	<i>Zai</i>
12.	س	<i>Sin</i>
13.	ش	<i>Syin</i>
14.	ص	<i>Shod</i>
15.	ض	<i>Dhod</i>
16.	ط	<i>Tho</i>
17.	ظ	<i>Zho</i>
18.	ع	<i>Ain</i>
19.	غ	<i>Ghain</i>
20.	ف	<i>Fa</i>
21.	ق	<i>Qaf</i>
22.	ك	<i>Kaf</i>
23.	ل	<i>Lam</i>
24.	م	<i>Mim</i>
25.	ن	<i>Nun</i>
26.	و	<i>Wau</i>
27.	ه	<i>Ha</i>

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Huruf Hijaiyah	Dibaca
28.	لا	<i>Lam Alif</i>
29.	ء	<i>Hamzah</i>
30.	ي	<i>Ya</i>

Sumber (Elyas, Ali Mustahib, dkk. 2011. Hal. 6-7)

Pada tabel di atas terlihat bahwa huruf hijaiyah yang tidak memakai tanda baca huruf seperti *fathah*, *domah*, *kasroh*, dan lain sebagainya dibaca sesuai dengan huruf aslinya.

3. *Harakat* (Tanda Baca Huruf)

Harakat merupakan bunyi huruf hijaiyah dalam Al-Qur'an yang dapat diubah – ubah sesuai dengan tanda bacanya. Berikut adalah gambar dari tanda baca huruf hijaiyah dalam Al-Qur'an yaitu :

a. *Fathah* : (َ)

Tabel 2. Tanda Baca *Fathah*

Menggunakan Tanda Baca <i>Fathah</i>														
أَ	بَ	تَ	ثَ	جَ	حَ	دَ	ذَ	رَ	زَ	سَ	شَ	صَ	ضَ	
طَ	ظَ	عَ	غَ	فَ	قَ	كَ	لَ	مَ	نَ	وَ	هَ	لَا	ءَ	يَ

(Sumber: Elyas, Ali Mustahib, dkk. *Pendidikan Agama Islam* : 2011. Hal. 8)

b. *Fathatain* : (ً)

Tabel 3. Tanda Baca *Fathatain*

Menggunakan Tanda Baca <i>Fathatain</i>														
أَ	بَ	تَ	ثَ	جَ	حَ	خَ	دَ	ذَ	رَ	زَ	سَ	شَ	صَ	ضَ
طَ	ظَ	عَ	غَ	فَ	قَ	كَ	لَ	مَ	نَ	وَ	هَ	لَ	ءَ	يَ

(Sumber : Elyas, Ali Mustahib, dkk. *Pendidikan Agama Islam* : 2011. Hal.10)

c. *Kasroh* : (ِ)

Tabel 4. Tanda Baca *Kasroh*

Menggunakan Tanda Baca <i>Kasroh</i>														
ا	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض
ط	ظ	ع	غ	ف	ق	ك	ل	م	ن	و	ه	لا	ء	ي

(Sumber : Elyas, Ali Mustahib, dkk. *Pendidikan Agama Islam* : 2011. Hal.9)

d. *Kasrotain* : (َ)

Tabel 5. Tanda Baca *Kasrotain*

Menggunakan Tanda Baca <i>Kasrotain</i>														
ضِ	صِ	شِ	سِ	زِ	رِ	ذِ	دِ	خِ	حِ	جِ	ثِ	تِ	بِ	اِ
يِ	ءِ	لِ	هِ	وِ	نِ	مِ	لِ	كِ	قِ	فِ	غِ	عِ	ظِ	طِ

(Sumber : Elyas, Ali Mustahib, dkk. *Pendidikan Agama Islam* : 2011. Hal. 10)

e. *Domah* : (ُ)

Tabel 6. Tanda Baca *Domah*

Menggunakan Tanda Baca <i>Domah</i>														
ضُ	صُ	شُ	سُ	زُ	رُ	ذُ	دُ	خُ	حُ	جُ	ثُ	تُ	بُ	أُ
يُ	ءُ	لُ	هُ	وُ	نُ	مُ	لُ	كُ	قُ	فُ	غُ	عُ	ظُ	طُ

(Sumber : Elyas, Ali Mustahib, dkk. *Pendidikan Agama Islam* : 2011. Hal. 9)

f. *Domatain* : (ً)

Tabel 7. Tanda Baca *Domatain*

Menggunakan Tanda Baca <i>Domatain</i>														
ضً	صً	شً	سً	زً	رً	ذً	دً	خً	حً	جً	ثً	تً	بً	أً
يً	ءً	لً	هً	وً	نً	مً	لً	كً	قً	فً	غً	عً	ظً	طً

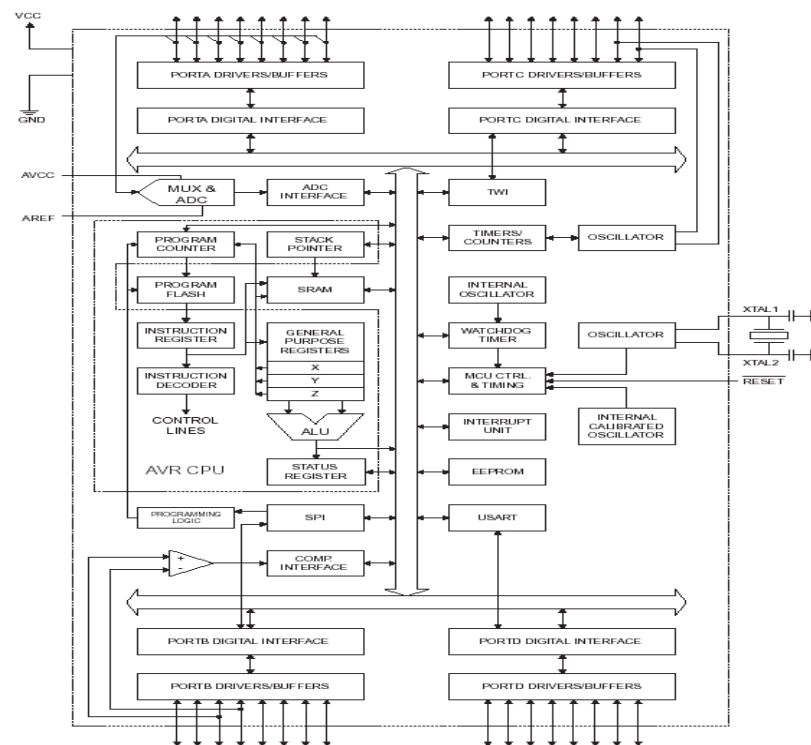
(Sumber : Elyas, Ali Mustahib, dkk. *Pendidikan Agama Islam* : 2011. Hal. 11)

D. Mikrokontroler ATmega16

Mikrokontroler AVR (*Alf and Vegard's Risc processor*) merupakan salah satu jenis arsitektur mikrokontroler yang menjadi andalan Atmel yaitu mikrokontroler RISC 8 bit berdasarkan arsitektur *Harvard*, yang dibuat oleh Atmel pada tahun 1996. AVR memiliki keunggulan dibandingkan dengan mikrokontroler lain, keunggulan AVR yaitu AVR memiliki kecepatan eksekusi program yang lebih cepat, karena sebagian besar instruksi dieksekusi dalam 1 siklus *clock*, lebih cepat dibandingkan MCS51 yang membutuhkan 12 siklus *clock* untuk mengeksekusi 1 instruksi.

1. Arsitektur Mikrokontroler ATmega16

Pada blok diagram ATmega16 di atas dapat dijelaskan beberapa bagian penting dari ATmega16 adalah sebagai berikut :



Gambar 1. Blok Diagram ATmega16

(Sumber : <http://www.atmel.com/Images/doc2466.pdf>)

- a. *Flash* adalah suatu jenis *ROM (Read Only Memory)* yang difungsikan untuk mengisi program hasil buatan manusia yang dijalankan oleh mikrokontroler.
- b. *RAM (Random Acces Memory)* merupakan memori yang membantu *CPU* untuk menyimpan data sementara dan pengolahan data ketika program sedang *running*.
- c. *EEPROM (Electrically Erasable Programmable Read Only Memory)* adalah memori untuk menyimpan data secara permanen oleh program yang sedang *running*.
- d. *Port I/O* adalah kaki untuk jalur keluar atau masuk sinyal sebagai hasil keluaran dan masukan bagi program.
- e. *Timer* adalah modul dalam *hardware* yang bekerja untuk menghitung waktu / pulsa.
- f. *PWM (Pulse Width Modulation)* adalah fasilitas untuk dapat membuat modulasi pulsa.
- g. *ADC (Analog to Digital Converter)* adalah fasilitas untuk dapat menerima sinyal analog dalam *range* tertentu untuk kemudian dikonversi menjadi suatu nilai digital dalam *range* tertentu.
- h. *SPI (Serial Peripheral Interface)* adalah jalur komunikasi data khusus secara *serial synchronous*.
- i. *ISP (In System Programming)* adalah kemampuan khusus mikrokontroler untuk dapat diprogram langsung dalam sistem rangkaiannya dengan membutuhkan jumlah *pin* minimal.

Menurut Solihul (2008) ATmega16 mempunyai *throughput* mendekati 1 MIPS membuat disainer sistem untuk mengoptimasi konsumsi daya versus kecepatan proses. Beberapa keistimewaan dari AVR ATmega16 antara lain :

a. *Advanced RISC Architecture*

- 1) *130 Powerful Instructions – Most Single Clock Cycle Execution.*
- 2) *32 x 8 General Purpose Fully Static Operation.*
- 3) *Up to 16 MIPS Throughput at 16 MHz.*
- 4) *On-chip 2-cycle Multiplier.*

b. *Nonvolatile Program and Data Memories*

- 1) *8 K Bytes of In-System Self-Programmable Flash.*
- 2) *Optional Boot Code Section with Independent Lock Bits.*
- 3) *512 Bytes EEPROM.*
- 4) *512 Bytes Internal SRAM.*
- 5) *Programming Lock for Software Security.*

c. *Peripheral Features*

- 1) *Two 8-bit Timer/Counters with Separate Prescalers and Compare Mode.*
- 2) *Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes.*
- 3) *One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode.*
- 4) *Real Time Counter with Separate Oscillator.*
- 5) *Four PWM Channels.*
- 6) *8-channel, 10-bit ADC.*

7) *Byte-oriented Two-wire Serial Interface.*

8) *Programmable Serial USART.*

d. *Special Microcontroller Features*

1) *Power-on Reset and Programmable Brown-out Detection.*

2) *Internal Calibrated RC Oscillator.*

3) *External and Internal Interrupt Sources.*

4) *Six Sleep Modes : Idle, ADC Noise Reduction, Power-save, power-down.*

5) *Standby and Extended Standby.*

e. *I/O and Package*

1) *32 Programmable I/O Lines.*

2) *40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad MLF.*

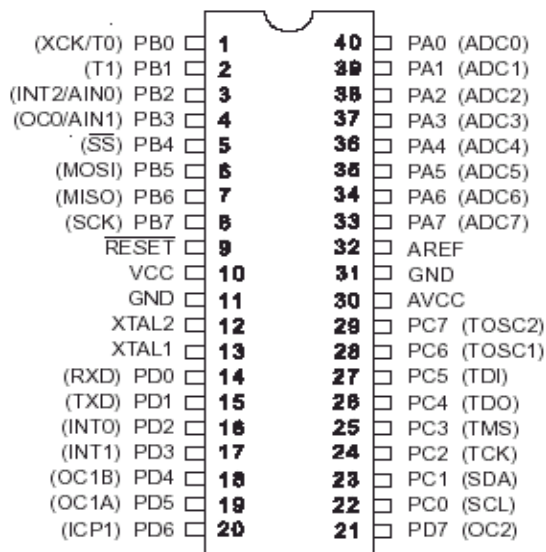
f. *Operating Voltages*

1) *2.7 - 5.5V for Atmega16L.*

2) *4.5- 5.5V for Atmega16.*

2. Konfigurasi ATmega16

Menurut Heri (2008) ATmega16 memiliki 40 pin, susunan pin mikrokontroler ATmega16 ini terdiri dari VCC, Ground, *port A*, *Port B*, *port C*, *port D*, *reset* dan lain sebagainya. Untuk lebih jelasnya perhatikanlah Gambar 2 di bawah ini.



Gambar 2. Konfigurasi Pin ATmega16
(Sumber :<http://www.atmel.com/Images/doc2466.pdf>)

Konfigurasi pin ATmega16 dengan 40 pin DIP (*dual in-line package*) dapat dilihat pada Gambar 2, dari Gambar di atas dapat dijelaskan fungsi dari masing – masing ATmega16 sebagai berikut :

- VCC : *Supply* tegangan *digital*.
- GND : *Ground*.
- Port A : *Port A* sebagai *input* analog ke A/D *converter*. *Port A* juga sebagai 8-bit *bi-directional Port I/O*, jika A/D *converter* tidak digunakan. Pin - pin *port* dapat menyediakan resistor - resistor internal *pull-up*. Ketika *port A* digunakan sebagai *input* dan pull eksternal yang rendah akan menjadi sumber arus jika resistor - resistor *pull-up* diaktifkan. Pin - pin *port A* adalah *tri-state* ketika kondisi *reset* menjadi aktif sekalipun *clock* tidak aktif.
- Port B : *Port B* adalah *Port I/O* 8-bit *bi-directional* dengan resistor

internal *pull-up*. *Buffer* output *port* B mempunyai karakteristik *drive* yang simetris dengan kemampuan keduanya *sink* dan *source* yang tinggi. Sebagai *input*, *port* B yang mempunyai *pull eksternal* yang rendah akan menjadi sumber arus jika resistor - resistor *pull-up* diaktifkan. Pin - pin *port* B adalah *tri-state* ketika kondisi *reset* menjadi aktif sekalipun *clock* tidak aktif.

e. *Port C* : *Port* C adalah *port* I/O 8-bit *bi-directional* dengan resistor - resistor *internal pull-up*. *Buffer* output *Port* C mempunyai karakteristik *drive* yang simetris dengan kemampuan keduanya *sink* dan *source* yang tinggi. Sebagai *input*, *port* C yang mempunyai *pull-eksternal* yang rendah akan menjadi sumber arus jika resistor - resistor *pull-up* diaktifkan. Pin - pin *port* C adalah *tri-state* ketika kondisi *reset* menjadi aktif sekalipun *clock* tidak aktif. Jika antarmuka JTAG *enable*, resistor - resistor *pull-up* pada pin - pin PC5 (TDI), PC3 (TMS), PC2 (TCK) akan diaktifkan sekalipun terjadi *reset*.

f. *Port D* : *Port* D adalah *port* I/O 8-bit *bi-directional* dengan resistor - resistor *internal pull-up*. *Buffer* output *port* D mempunyai karakteristik *drive* yang simetris dengan kemampuan keduanya *sink* dan *source* yang tinggi. Sebagai *input*, *port* D yang mempunyai *pull eksternal* yang rendah akan menjadi sumber arus jika resistor -

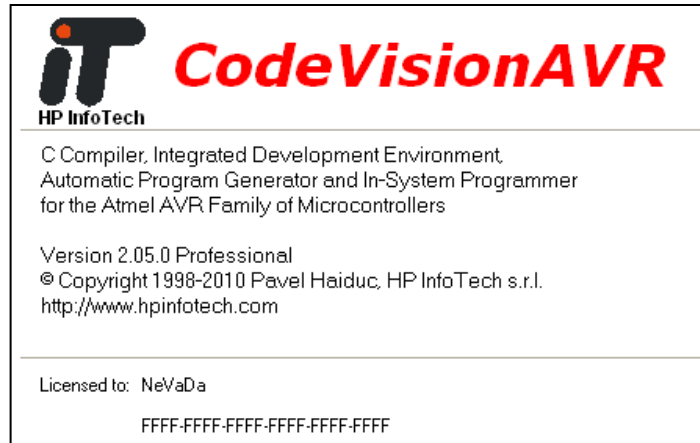
resistor *pull-up* diaktifkan. Pin-pin *port D* adalah *tri-state* ketika kondisi *reset* menjadi aktif sekli pun *clock* tidak aktif.

- g. *Reset* : Sebuah *low level* pulsa yang lebih lama dari pada lebar pulsa minimum pada pin ini akan menghasilkan *reset* meskipun *clock* tidak berjalan.
- h. *XTAL1* : *Input inverting* penguat Oscilator dan *input* intenal *clock* operasi rangkaian.
- i. *XTAL2* : *Output* dari *inverting* penguat Oscilator.
- j. *AVCC* : Pin tegangan *supply* untuk *port A* dan *A/D converter*.
Sebaiknya eksternalnya dihubungkan ke *VCC* meskipun *ADC* tidak digunakan. Jika *ADC* digunakan seharusnya dihubungkan ke *VCC* melalui *low pass filter*.
- k. *AREF* : Pin tegangan referensi untuk *A/D converter*.

E. CodeVision AVR version 2.05.0

Menurut Heri (2008) *codevision AVR* merupakan sebuah *cross-compiler C*, *IDE (Integrated Development Environtment)*, di mana penulisan program, *compile*, *link*, pembuatan kode mesin (*Assembler*) dan program yang sudah *desain* dapat di-*download* ke *chip AVR* dengan menggunakan *codevision*, selain itu fasilitas yang melakukan komunikasi *serial* dengan mikrokontroler yang sudah diprogram. Pada proses pemindahan program ke IC mikrokontroler dapat dilakukan dengan *system download* secara *ISP (In-System Programming)*, *in-System Programmable Flash on-chip* membiarkan memori program untuk

diprogram ulang menggunakan hubungan *serial* SPI. Pada proses awal program CVAVR terlihat pada Gambar 3 sebagai berikut.



Gambar 3. *CodeVision AVR version 2.05.0*

CodeVision AVR memiliki beberapa aplikasi yang dilengkapi dengan *source code editor*, *compiler*, *linker* dan dapat memanggil Atmel AVR Studio untuk *debugger*-nya. Pada penggunaan *Code Wizard* dari tampilan konfirmasi memberikan perintah untuk membuat *Project* baru, sehingga menampilkan beberapa tampilan konfigurasi USART, *Analog Comparator*, ADC, SPI, I2C, 1 Wire (I2C), LCD, *Bit-Banged*, *Project Information*, *Chip*, *Port*, *External IRQ*, *Timer*. Program yang dibuat melalui *code Wizard* dapat mengatur program. Misalnya untuk konfigurasi *chip* yang akan digunakan berupa : Atmega16, *Clock* : 11.059200 MHz dan pengaturan *port* sebagai *input* dan *output*.

F. *Proteus ISIS* dan *ARES*

Menurut Syaifudin (2011) *ISIS* adalah suatu *software* yang memiliki kemampuan untuk mensimulasi terhadap *firmware* pada *embedded* sistem seperti mikrokontroler. Dengan *proteus* terinstal dikomputer kita, maka seakan - akan kita telah memiliki sebuah labolatorium elektronika analog maupun digital. Fitur

– fitur ini berupa komponen elektronika, alat ukur (*oscilloscope, frequency counter, spectrum analyzer, multimeter*, dan lain -lain). Dengan adanya *software* ini dapat dijadikan sebagai labolatorium elektronika untuk mempermudah dalam membuat rangkaian elektronika.

ARES merupakan *software* yang digunakan untuk mendesain *layout* PCB. *Software* ini merupakan salah satu modul dari *proteus*. Sistem pada *software* ini dilengkapi perangkat untuk mendesain PCB dengan beragam *tool* yang disediakan.

G. Keyboard

Menurut Zaenal Muttaqien (2012) *keyboard* merupakan perintah *input* berbentuk papan ketik yang memiliki tombol huruf, tombol angka, tombol karakter khusus dan tombol fungsi. Keyboard dapat berfungsi sebagai media bagi pemakai untuk melakukan perintah - perintah lainnya seperti menyimpan dan membuka file data yang sudah disimpan.

Keyboard memiliki konektor memiliki 6 buah terminal yang digunakan untuk menjalankan keyboard sesuai dengan fungsinya, namun terminal yang dipakai terdiri dari 4 buah yaitu terminal VCC, *ground*, *clock*, dan data.

Berikut ini adalah gambar keyboard PS2 6 konektor yang digunakan untuk mengirim tombol huruf, angka atau karakter kedalam sebuah PC. Berikut adalah Gambar keyboard PS2 yang memiliki 6 konektor.



Gambar 4. *Keyboard PS2 6 connector*

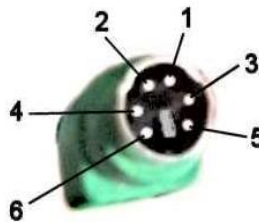
(Sumber : <http://aksesoris-comp.blogspot.com/2012/05/keyboard.html>)

Setiap kali salah satu tombol *keyboard* ditekan atau dilepas, *keyboard* akan mengirim kode ke *host* yang dihubungkan ke mikrokontroler. Kode tersebut dinamakan sebagai *scan code*. *Scan Code* tombol ‘S’ adalah 1B (angka *heksa decimal* setara dengan angka *biner* 00011011). Ketika tombol ‘S’ ditekan terus maka *keyboard* akan mengirimkan 1B berikutnya secara terus menerus, sampai ada tombol lain yang ditekan atau tombol ‘S’ dilepas. *Keyboard* juga mengirim kode saat ada satu tombol yang dilepas, kodenya adalah F0 (angka *heksa desimal* setara dengan angka *biner* 11110000), jadi kalau tombol ‘S’ tadi dilepas, *keyboard* akan mengirim F0 dan 1B. Kode-kode tersebut dikirim *keyboard* secara seri, artinya dikirimkan satu *bit* demi satu *bit*. Misalnya 1B dikirimkan dengan cara : mula - mula kirim ‘1’, kemudian ‘1’ lagi dan menyusul ‘0’ sampai akhirnya terkirim sebanyak 8 bit yang terbentuk 00011011 (dimulai dari *bit* yang paling kanan kemudian bergeser satu persatu sampai yang paling kiri). Setiap tombol *keyboard* memiliki *scan code* yang berbeda-beda. Pada gambar di bawah ini adalah Gambar *Scan Code* pada *keyboard*.

Esc	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	Print	Scroll	Break	1" x 8412507C				
76	05	06	04	0C	03	0B	03	0A	01	09	7B	07	*1	7E	*2	2" x 87477E 1F04F077				
OE	16	1E	25	26	2E	36	3D	3E	46	45	4E	55	65	del	home	end	77	004A	7C	7B
TAB	Q	W	E	R	T	Y	U	I	O	P	[]	del	home	end	77	004A	7C	7B	
00	15	1D	24	2D	2C	35	3C	43	44	4D	54	5D	del	home	end	77	004A	7C	7B	
Cap	A	S	D	F	G	H	J	K	L	;	'	5A	del	home	end	77	004A	7C	7B	
58	1C	1B	23	2B	34	33	3B	42	4B	4C	52	5A	del	home	end	77	004A	7C	7B	
Shift	Z	X	C	V	B	N	M	,	.	/	Shift	5A	del	home	end	77	004A	7C	7B	
12	1A	22	21	2A	32	31	34	41	49	4A	59	5A	del	home	end	77	004A	7C	7B	
Ctrl	Alt		Space							Alt	Ctrl		del	home	end	77	004A	7C	7B	
14	11		29							Alt	Ctrl		del	home	end	77	004A	7C	7B	

Gambar 5. Scan code masing-masing tombol keyboard PC
(Sumber : <http://www.mytutorialcafe.com/mikrokontroller%20keyboard.htm>)

Di bawah ini adalah gambar konektor dari keyboard yang digunakan sebagai penghubung input dari tiap code tombol keyboard.



Gambar 6. Konektor keyboard PS2 (MINI-DIN6)
(Sumber: www.mp3car.com)

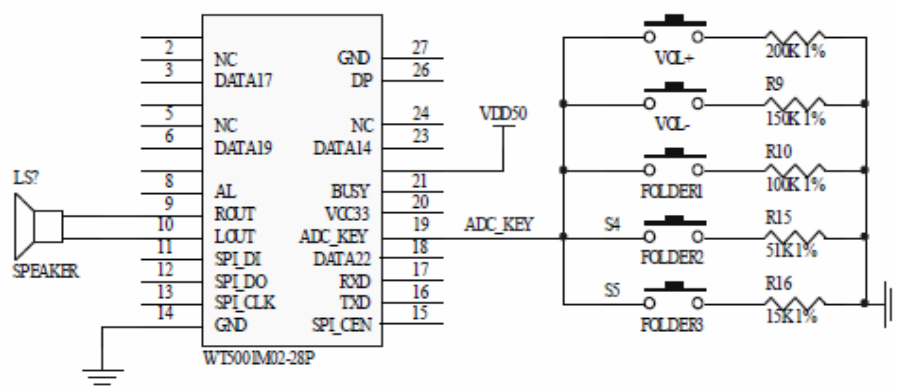
Dari gambar tersebut dapat dijelaskan bahwa pada masing - masing pin connector PS2 Female ini memiliki beberapa fungsinya yaitu:

1. Pin nomor 1 adalah KBD clock berfungsi sebagai clock pengiriman masing - masing bit data.
2. Pin nomor 2 adalah N/C not connected.
3. Pin nomor 3 adalah grounding.
4. Pin nomor 4 adalah VCC sebagai catu daya 5V DC.
5. Pin nomor 5 adalah KBD data sebagai output data keyboard.
6. Pin nomor 6 adalah N/C not connected.

H. Modul MP3 Player

WT5001 - 48L adalah sebuah *chip* suara MP3 berkualitas tinggi dari OTP 8 BIT CISC, mendukung MP3 decoding, mendukung format WAV, dengan seperangkat perangkat antarmuka, SPI, mendukung DMA, drive 16 Ω headphone, dengan jam *real-time* dan modul jam kalender, standar RS232 seri pengendalian. Dual channel LDO, 3.3V- 5.0V, 1.8V-3.3V, daya rendah kesempatan, siaga 60 μ saat Ampere, respon yang cepat saat mengendalikan.

WT5001M01 - 16P, WT5001M02 dan WT5001M03-28P termasuk ke dalam modul MP3 Player berkualitas tinggi dan stabil. Modul setengah jadi dikembangkan sebagai WT5001-48L adalah *chip* utama. Dari ke 3 modul ini memiliki fungsi sistem minimum WT5001-48L dengan SPI-FLASH sebagai media *stronge*, 1W amplifier, fungsi mengendalikan *portserial*, *file download music online*. Berdasarkan hal ini, masing - masing memiliki *fitur* yang berbeda, dan *fleksibel* dengan perkembangan sekunder dalam berbagai situasi dan dapat digunakan secara langsung. Berikut ini adalah Gambar diagram minimum rangkaian aplikasi tipe WT5001M02 - 28P. <http://waytronic.ru/waytronic15.htm>



Gambar 7. WT5001M02-28P Minimum
(Sumber : <http://waytronic.ru/waytronic15.htm>)

I. *Micro SD Card (Micro-Secure Digital Card)*

Micro SD Card yaitu sebuah kartu memori tempat penyimpanan (*non-volatile*) yang dikembangkan sebagai atas *kompatibel* ke *multi media card*. Di dalam *Micro SD* mikrokontroler yang terdiri dari memori *flash* kontrol yang digunakan untuk menghapus, membaca, menulis dan mengontrol kesalahan. Data ditransfer antara kartu memori dan *host controller* sebagai data blok di unit 512 *byte*.

1. Sistem Penulisan *Micro SD*

Micro SD mempekerjakan *NAND Flash Memory* sebagai *array* memori. Memori *flash NAND* adalah nilai efektif dan dapat membaca/menulis data yang lebih besar serta cepat dalam pemrosesannya, tapi di sisi lain, ada kelemahan yang menulis ulang bagian dari data yang tidak efisien. Umumnya memori *flash* membutuhkan untuk menghapus data yang ada sebelum menulis data baru, dan satuan minimum operasi menghapus (disebut menghapus blok) lebih besar dari ukuran menulis blok. Memori *flash NAND* khas memiliki ukuran blok 512/16K *byte* untuk operasi menulis/menghapus, dan kartu ini berukuran lebih besar mempekerjakan blok *chip* besar (2K/128K). Ini berarti bahwa menulis ulang seluruh data dalam blok menghapus dilakukan dalam kartu bahkan jika menulis hanya sektor (512 *byte*).

2. Sistem Penyimpanan *Micro SD*

Micro SD sudah terformat dengan sistem *file* sebagai FAT16, SDHC sebagai FAT32, sedangkan SDXC sebagai ExFAT. Pada FAT16 dan FAT32

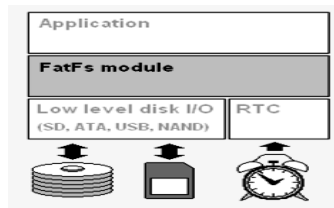
memungkinkan untuk dapat diakses melalui semua perangkat host untuk pembaca *SD*. Pemeliharaan FAT dengan standar dapat digunakan untuk memperbaiki atau mengambil data yang rusak dan beberapa dapat memulihkan *file* yang dihapus. Namun karena teknologi ini muncul sebagai *drive removable hard*, sehingga bisa diformat ulang untuk setiap sistem *file* yang didukung oleh sistem operasi. Juga dapat dilakukan dengan sistem operasi seperti *USB Live* yang bisa memulihkan *host* komputer dari *Flash Media Reader*.

3. Mengakses *File MP3* Pada *SD Card* dengan Mikrokontroler

Selain untuk penyimpanan data melalui antarmuka langsung menggunakan komputer, *SD Card* dapat diakses melalui mikrokontroler, khusus untuk *file* dengan tipe *mp3* dan beberapa tipe lainnya. Pada perangkat lunak *Codevision AVR* telah dilengkapi dengan *library* untuk mengakses *SD Card*. *Library* yang terdapat dalam *Codevision AVR* berupa modul/fungsi *FatFS*.

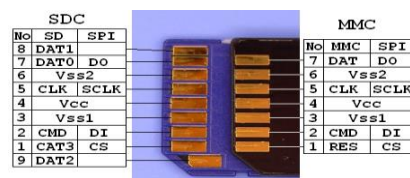
FatFs merupakan modul/fungsi untuk membaca sistem *file FAT*. *FatFs* memiliki standar ANSI (*American National Standart Institute*) dalam pemrograman bahasa C dan sepenuhnya terpisah dari bagian *input/output*. Oleh karena modul ini tidak tergantung pada arsitektur perangkat keras. Hal ini dapat dimasukkan ke dalam mikrokontroler seperti AVR, PIC 8051, ARM, Z80, 68k dan lain-lain. <http://www.elm-chan.org>.

Berikut adalah Gambar dari ilustrasi pengaksesan level minimum *Disk I/O* pada modul *FatFs*.



Gambar 8. Ilustrasi Pengaksesan *Low Level Disk I/O* pada Modul *FatFs*
(Sumber : <http://www.elm-chan.org>)

Pada Gambar di atas menjelaskan tentang cara mengakses aplikasi (mikrokontroler) dengan perangkat *I/O disk* tingkat rendah seperti *SD Card* dan sebagainya. Gambar ini mengilustrasikan bahwa Modul *FatFs* berfungsi sebagai perantara antara aplikasi (mikrokontroler) dengan perangkat *input/output (SD Card)*. Sistem antarmuka antara mikrokontroler dan *SD Card* menggunakan teknik antarmuka SPI. Koneksi antara *SD Card* dengan Mikrokontroler dengan sistem SPI terlihat pada Gambar 9.



Gambar 9. Koneksi *Pin* Untuk Antarmuka *SD Card/MMC* dengan Mikrokontroler (Sistem SPI)
(Sumber : <http://www.elm-chan.org>)

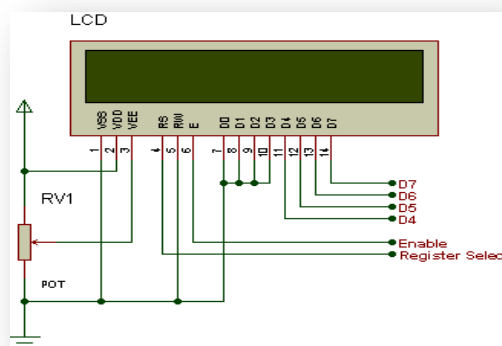
Data *file* pada tipe *mp3* memiliki keuntungan dibandingkan dengan data *file* yang lainnya. Keuntungan dari tipe data *mp3* memakai kapasitas memori lebih sedikit dibandingkan dengan *file* tipe *wav* dan *midi*. Namun dalam penggunaannya tipe *file* dari segi suara yang kurang jernih dikarenakan *file* tipe *mp3* ini memakan memori sedikit dibandingkan *file* tipe yang lainnya. Dalam perancangan media pembelajaran, penulis memilih *file* dengan tipe *mp3* karena kemudahannya jika diakses dengan pemrograman mikrokontroler. Meskipun memiliki kualitas suara

yang kurang jernih, akan tetapi *file type mp3* sudah cukup baik bila digunakan sebagai *output* suara pada alat media pembelajaran.

J. LCD (*Liquid Crystal Display*) 16 x 2

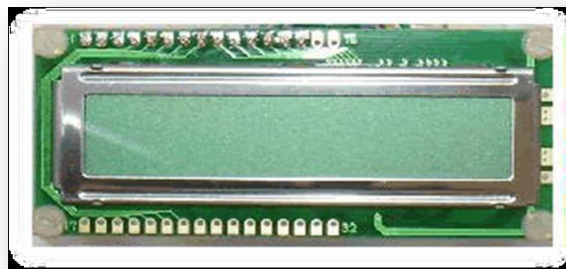
LCD (Liquid Crystal Display) merupakan media penampil elektronik terbuat dari bahan cairan *Cristal* yang digunakan untuk membentuk karakter dan sistem pengoperasiannya menggunakan sistem dot matriks. LCD ini banyak diminati karena harganya yang ekonomis, mudah diprogram, tidak memiliki batasan menampilkan khusus serta banyak digunakan sebagai *display* peralatan elektronik berupa jam digital, alat pengukur digital dan lain sebagainya.

Sebuah LCD 16x2 berarti dapat menampilkan 16 karakter per baris dan ada 2 jalur tersebut. Dalam LCD ini masing-masing karakter ditampilkan dalam matriks 5x7 pixel. LCD ini memiliki dua *register* yaitu : *Command* dan Data. *ASCII* dari karakter yang akan ditampilkan pada LCD. Klik untuk mempelajari lebih lanjut tentang struktur *internal* LCD. <http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet>.



Gambar 10. Blok Diagram LCD

Pada Gambar LDC Karakter 2x16 di atas biasa digunakan untuk memberikan informasi pemilihan simulasi, saat salah satu simulasi telah dipilih, LCD akan menampilkan tulisan jenis kerusakan pada *line* 1 dan menampilkan tegangan kerja normal dibandingkan dengan tegangan kerja setelah di *Trouble* pada *line* 2. Berikut contoh gambar LCD yang memiliki karakter 16x2.



Gambar 11. LCD karakter 16 x 2
(Sumber :<http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet>)

Perintah adalah sebuah instruksi yang diberikan kepada LCD untuk melakukan tugas yang telah ditetapkan seperti pada waktu memulainya, membersihkan layarnya, pengaturan posisi kursor, mengendalikan layar. Perintah dapat dilakukan dengan cara menyimpan instruksi yang diberikan ke LCD.

LCD dapat dengan mudah di hubungkan dengan ATmega16 karena tidak perlu membutuhkan *driver* atau rangkaian lain untuk menghubungkannya. Untuk menghubungkan *port* yang ada pada IC ATmega16, bisa langsung dihubungkan dengan kaki-kaki pada LCD. Hal yang paling penting dan harus selalu diingat yaitu ketika menghubungkan *port* pada IC ATmega16 dengan LCD ialah harus memperhatikan kaki-kaki nomor berapa saja yang kita gunakan pada LCD agar

tampilan sesuai dengan yang di harapkan. Di bawah ini merupakan Tabel *pin* LCD dan fungsinya :

Tabel 8. Pin LCD dan Fungsinya

PIN	Nama PIN	Fungsi
1	VSS	Ground voltage
2	VCC	+5V
3	VEE	Contrast voltage
4	RS	Register Select 0 = Intruction Register 1 = Data Register
5	R/W	Read/Write, to choose write or read mode 0 = Write mode 1 = Read mode
6	E	Enable 0 = Start to lacht data to LCD character 1 = Disable
7	DB0	Data bit ke-0 (LSB))
8	DB1	Data bit ke-1
9	DB2	Data bit ke-2
10	DB3	Data bit ke-3
11	DB4	Data bit ke-4
12	DB5	Data bit ke-5
13	DB6	Data bit ke-6
14	DB7	Data bit ke-7 (MSB)
15	BPL	Back Plane Light
16	GND	Ground

K. IC Penguat Suara TDA7052

IC penguat suara merupakan komponen yang berfungsi untuk menguatkan suara. Suara dari suatu perangkat dikuatkan sesuai dengan penguatan untuk dikeluarkan melalui output *speaker*. Banyak macam IC penguat suara yang terdapat di pasaran mulai dari IC dengan daya rendah hingga daya penguatan yang tinggi. Salah satu IC dengan seri TDA7052 yang memiliki kualitas yang baik serta hemat energi. <http://www.alldatasheet.com>.

Berikut adalah bentuk fisik dari IC TDA 7052 yang terdiri dari 8 buah pin, terlihat seperti pada Gambar 12 di bawah ini.



Gambar 12. IC TDA7052

(Sumber : <http://www.voti.nl/shop/catalog.html?IC-TDA7052-DIP>)

Karakteristik IC TDA7052 adalah terdiri dari beberapa tingkat penguat. Penguat dari IC TDA7052 menggunakan sebuah penguat OP-AMP serta 2 buah *buffer*.

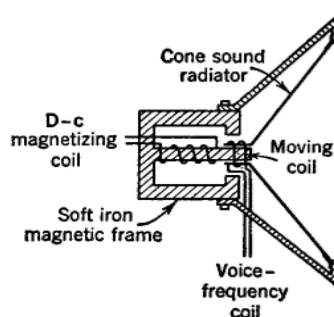
IC TDA7052 merupakan IC penguat suara tunggal kanal (mono). Dengan jumlah kaki sebanyak 8 pin, IC TDA7052 praktis digunakan untuk kebutuhan *amplifier* yang minimalis. IC TDA7052 dapat bekerja pada tegangan 3-15 Volt. Keunggulan IC ini dibandingkan dengan IC atau penguat lain adalah hemat energi, tidak memerlukan pendingin dalam pemasangannya, serta tidak memerlukan komponen yang banyak dalam pemasangan. Dari keunggulan serta

konstruksi yang minimalis sehingga banyak yang memilih IC TDA7052 sebagai penguat suara dalam rancangan media pembelajaran huruf hijaiyah.

L. *Speaker*

Menurut Wayne Storr (2013) *speaker* merupakan *transduser* yang mengubah sinyal elektrik menjadi suara. Pada dasarnya *speaker* merupakan mesin penerjemah akhir dalam sistem *audio*. Kebalikan dari *mikrofon* yang memiliki prinsip mengubah getaran suara menjadi sinyal elektrik, *speaker* membawa sinyal elektrik (yang telah dikuatkan) dan mengubahnya kembali menjadi getaran untuk membuat gelombang suara yang dapat didengar.

Prinsip kerja *speaker* tidak lepas dari fungsi masing - masing bagian penampang *speaker* yang terlihat seperti Gambar 13. Dalam sistem *audio*, terdapat penguat atau *driver* yang berfungsi untuk memberi sinyal atau menggerakkan *speaker* agar dapat mengeluarkan suara. *Driver* ini dapat menggerakkan satu *speaker* atau lebih. Prinsip kerja *speaker* tidak lepas dari fungsi setiap bagian *speaker*.



Gambar 13. Penampang *Speaker*

(Sumber : <http://www.vias.org>)

Pada prinsip dari operasi *speaker* tersebut adalah kebalikan dengan yang ada pada *dynamic microphone*. Saat sebuah sinyal analog melewati kumparan suara atau *speaker*, medan elektro magnetik yang dihasilkan ditentukan oleh besarnya arus yang mengalir pada kumparan suara. Pada besarnya arusnya yang mengalir ditentukan oleh pengaturan *volume control amplifier*. Gaya elektro magnetik yang dihasilkan oleh bidang ini bertolak disekitar medan magnet permanen utama dan mencoba untuk mendorong kumparan dalam satu arah.

BAB III

KONSEP RANCANGAN

Perancangan Media Pembelajaran Huruf Hijaiyah dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16 ini terdiri dari proses observasi awal, identifikasi kebutuhan, analisis kebutuhan, blok diagram rangkaian, perancangan perangkat keras, perancangan perangkat lunak, pembuatan alat dan pengujian.

A. Identifikasi Kebutuhan

Identifikasi kebutuhan yang diperlukan dari pembuatan proyek akhir ini diantaranya adalah:

1. Blok atau skema rangkaian huruf hijaiyah dengan output suara berbasis mikrokontroler.
2. Perangkat keras sistem minimum mikrokontroler ATmega16 sebagai sistem kendali peralatan dan program.
3. Perangkat keras berupa tombol *input* untuk mengeksekusi program huruf hijaiyah.
4. Komponen yang digunakan untuk menyimpan dan memanggil *file* suara melalui pemrograman mikrokontroler.
5. Komponen untuk menguatkan suara dari keluaran mikrokontroler dan dapat menghasilkan suara sehingga terdengar oleh manusia.
6. Perangkat lunak yang dapat digunakan untuk melakukan pemrograman mikrokontroler.

7. Perangkat lunak untuk membuat desain PCB berdasarkan skema rangkaian

B. Analisis Kebutuhan

Berdasarkan identifikasi kebutuhan dapat diperoleh beberapa analisis kebutuhan sebagai berikut:

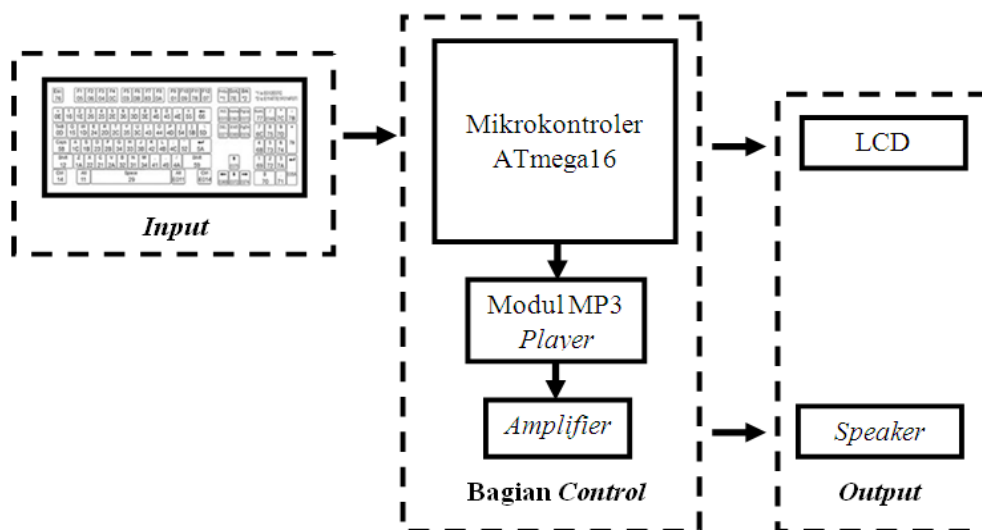
1. Blok rangkaian huruf hijaiyah menggunakan tombol *input* tekan, IC mikrokontroler, modul pemanggil *file* suara, perangkat penyimpan *file* suara dan rangkaian penguat suara.
2. Sistem minimum mikrokontroler ATmega16 digunakan untuk menyimpan program huruf hijayah.
3. Tombol tekan huruf hijaiyah menggunakan tombol *keyboard* sebagai *input* untuk mengeksekusi program. Tombol *keyboard* yang digunakan terdiri dari 30 tombol huruf hijaiyah dan 6 tombol tanda baca huruf.
4. Komponen penyimpan *file* suara menggunakan *micro sd* dengan *format file MP3* dan modul *mp3 player* sebagai media untuk memanggil *file* suara yang ada *micro sd*. *File* suara dapat dipanggil melalui IC mikrokontroler menggunakan komunikasi SPI.
5. Penguat suara keluaran dari mikrokontroler dapat menggunakan IC TDA 7052. Pada penguat suara yang digunakan ini memiliki tegangan rendah dan keluaran langsung dapat dihubungkan dengan *speaker*.
6. Pengisian program menggunakan perangkat *USBasp downloader* melalui *software CVAVR programmer* dengan media *interface* antara PC dengan IC mikrokontroler.

7. Perangkat lunak *Proteus* dapat digunakan untuk membuat desain rangkaian, simulasi rangkaian (*fitur ISIS*), sekaligus membuat desain PCB (*fitur ARES*).

C. Perancangan Sistem

1. Perancangan Blok Diagram Rangkaian

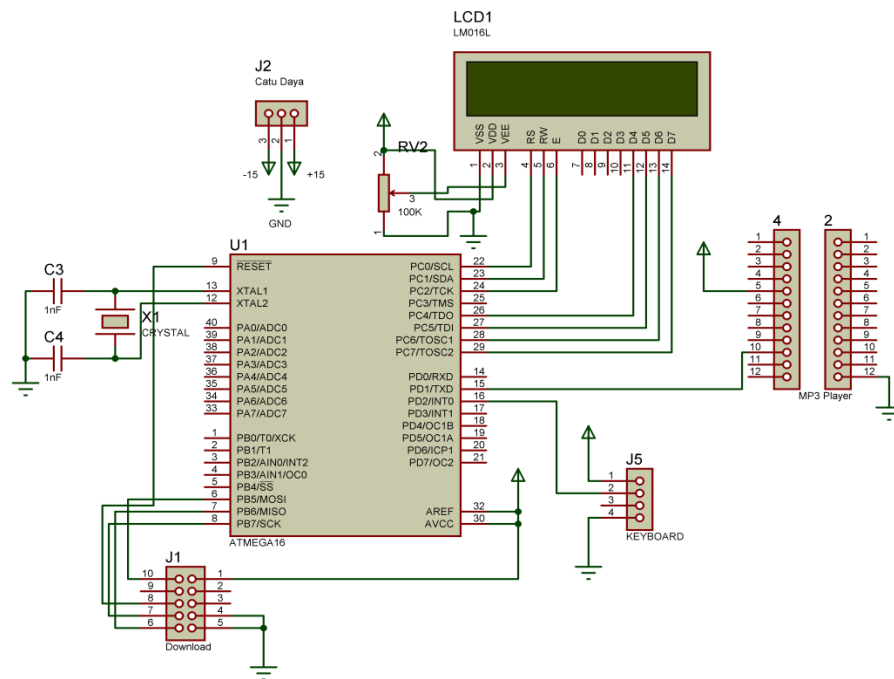
Blok diagram rangkaian dari pembuatan Media Pembelajaran Huruf Hijaiyah dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16 terdiri dari perangkat masukan, perangkat pemrosesan, dan perangkat keluaran. Gambar dibawah ini merupakan blok diagram rangkaian dalam rancangan huruf hijaiyah.



Gambar 14. Blok Diagram Sistem

2. Rangkaian Sistem Minimum ATmega16

Sistem minimum ATmega16 yaitu rangkaian yang berfungsi sebagai pengendali dan penghasil dari rangkaian suatu sistem. Di bawah ini adalah Gambar sistem minimum ATmega16.

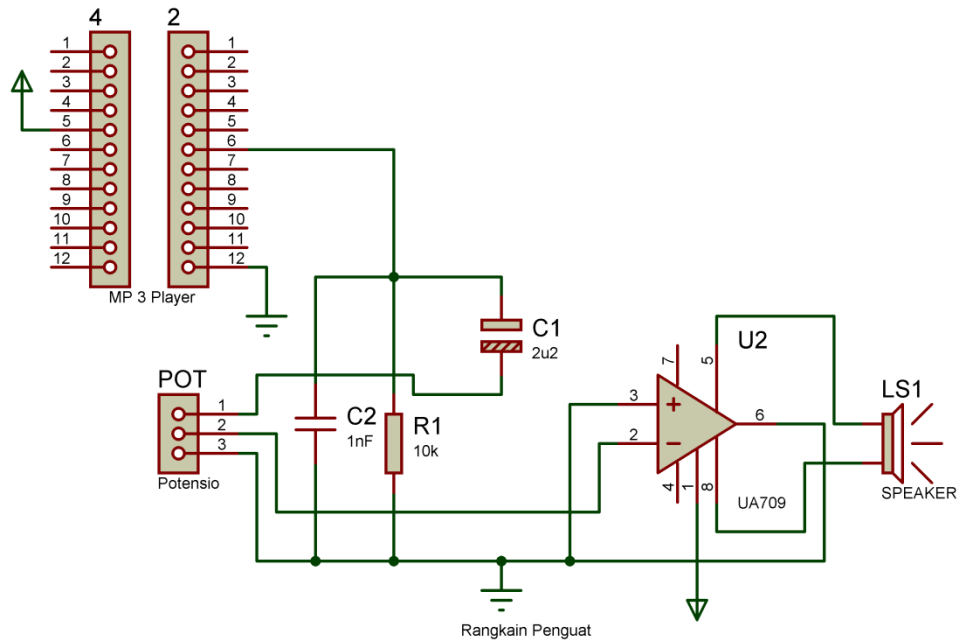


Gambar 15. Rangkaian Sistem Minimum

Rangkaian ini merupakan rangkaian sistem minimum yang digunakan sebagai pemroses data. Pada IC mikrokontroler ATmega16 ini sebagai pengendali utama dari keseluruhan sistem, berupa masukan dan keluaran dari sebuah program.

3. Rangkaian Modul *MP3 Player* dan *mini power Amplifier*

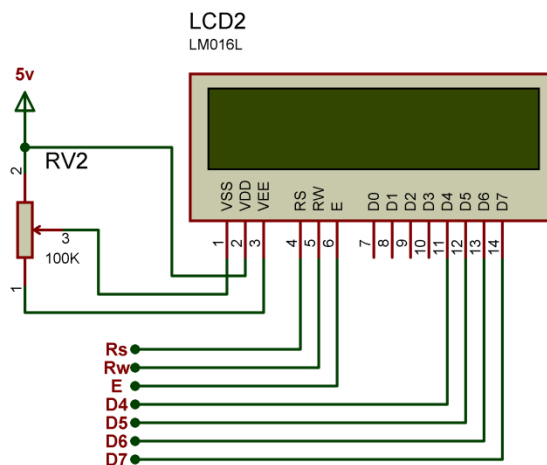
Pada rangkaian ini merupakan rangkaian yang berfungsi sebagai keluaran berupa output suara. Suara dihasilkan dari *file* suara yang tersimpan di dalam *micro sd*. Pada rangkaian ini menggunakan komponen modul *mp3 player* dan IC TDA 7052. Dibawah ini adalah rangkaian yang menghasilkan *output* suara.



Gambar 16. Rangkaian Modul *MP3 player* dan *mini power amplifier*

4. Rangkaian *Display LCD*

LCD berfungsi untuk menampilkan hasil keluaran berupa karakter dengan ukuran 16 x 2 karakter. *Port* pada LCD ini dihubungkan dengan *port* ATmega16. Berikut adalah Gambar rangkaian *display LCD* 16 x 2.



Gambar 17. Rangkaian *Display LCD* 16 x 2

Display LCD ini digunakan untuk menampilkan karakter, dapat dilihat pada rangkaian terhadap komponen *variable resistor* yang berfungsi mengatur kontras, dipilih 100K Ω merupakan *recommended* yang digunakan untuk rangkaian LCD 16 x 2 dengan *supply* tegangan yang diperlukan yaitu 5 volt.

5. Perancangan Perangkat Lunak

Perancangan perangkat lunak ini berisi tentang deskripsi yang berisi tentang langkah kerja program yang dibuat secara sistematis sesuai dengan rancangan. Pembuatan perangkat lunak diawali dengan pola algoritma dan diagram alir (*flowchart*) yang akan direlisasikan kedalam sebuah program. Kemudian program akan di *download* ke dalam sebuah *flash memory* pada ATmega16. Berikut ini adalah urutan dari perancangan perangkat lunak yang terdiri dari :

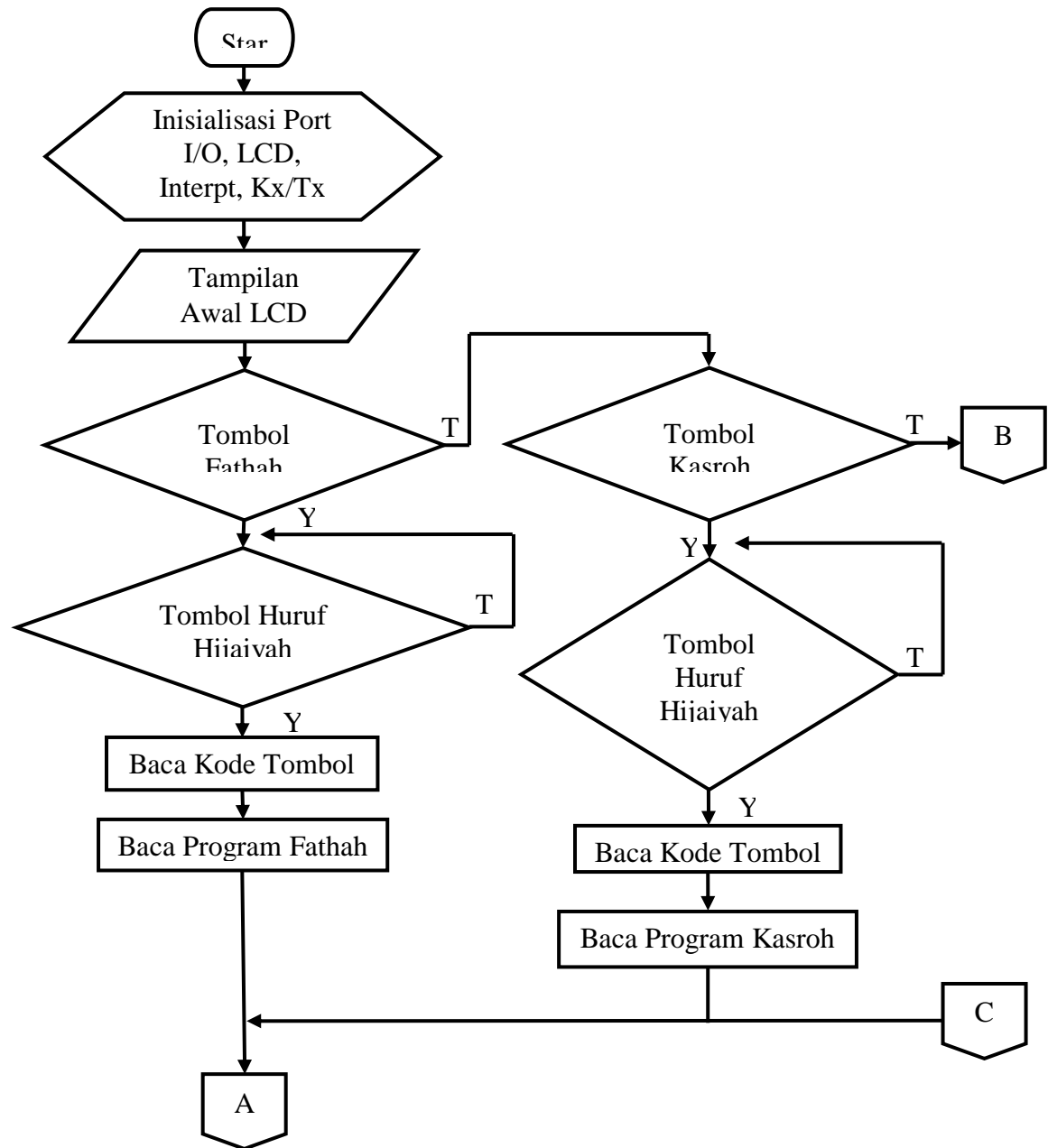
a. Algoritma

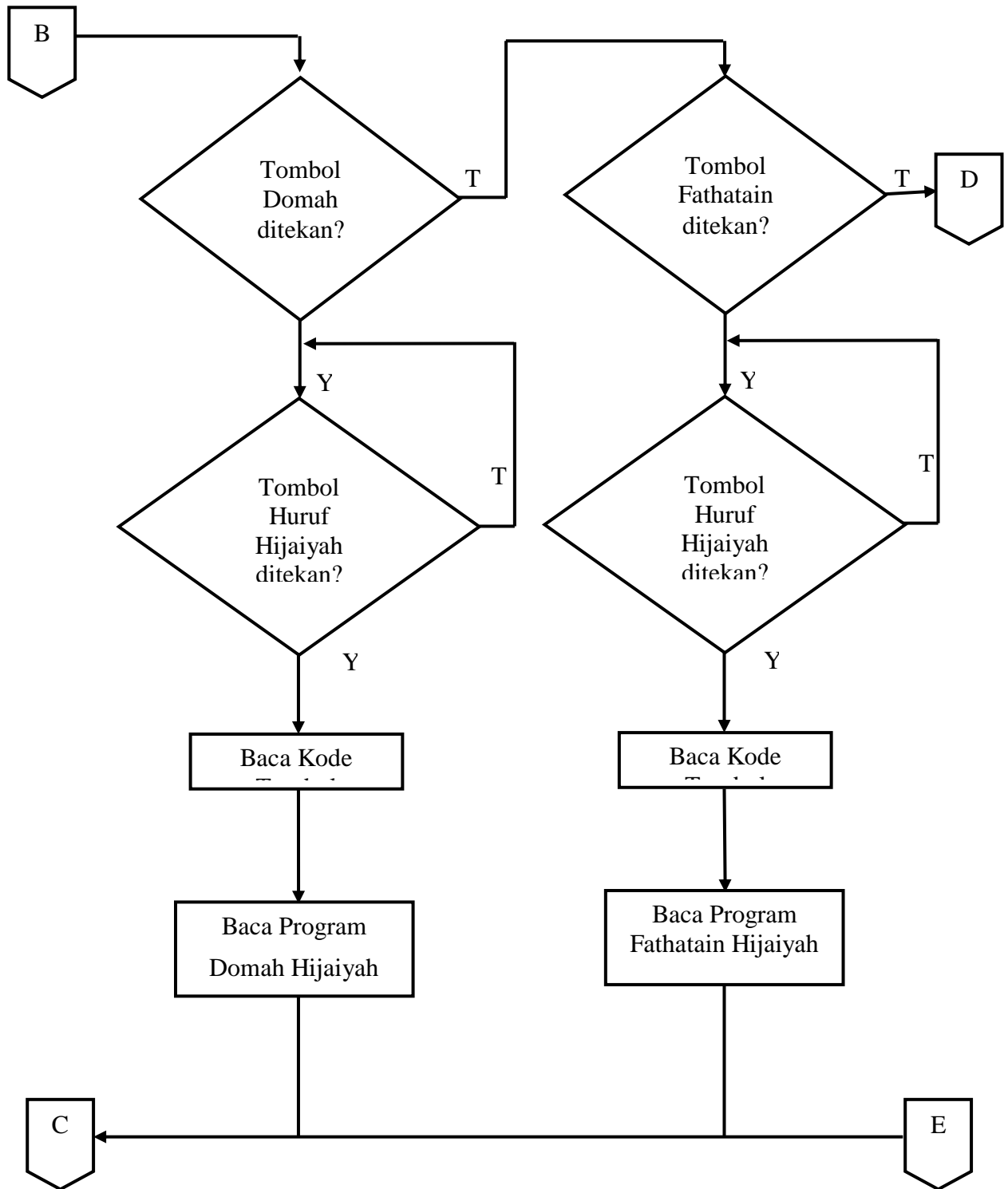
- Step 1* : Mulai.
- Step 2* : Inisialisasi PortI/O, LCD, Interrupt, Rx/Tx.
- Step 3* : Inisialisasi Delay.
- Step 4* : Tampilan awal display LCD.
- Step 5* : Apakah tombol huruf Fathah ditekan? Jika Ya maka lanjut ke Step 6, Jika tidak maka lanjut ke step 9.
- Step 6* : Apakah tombol huruf hijaiyah ditekan? Jika Ya maka lanjut step 7, Jika tidak kembali ke step sebelumnya.
- Step 7* : Baca Kode Tombol.
- Step 8* : Output LCD menampilkan bacaan huruf dan *Speaker* akan mengeluarkan suara dari huruf hijaiyah fathah.

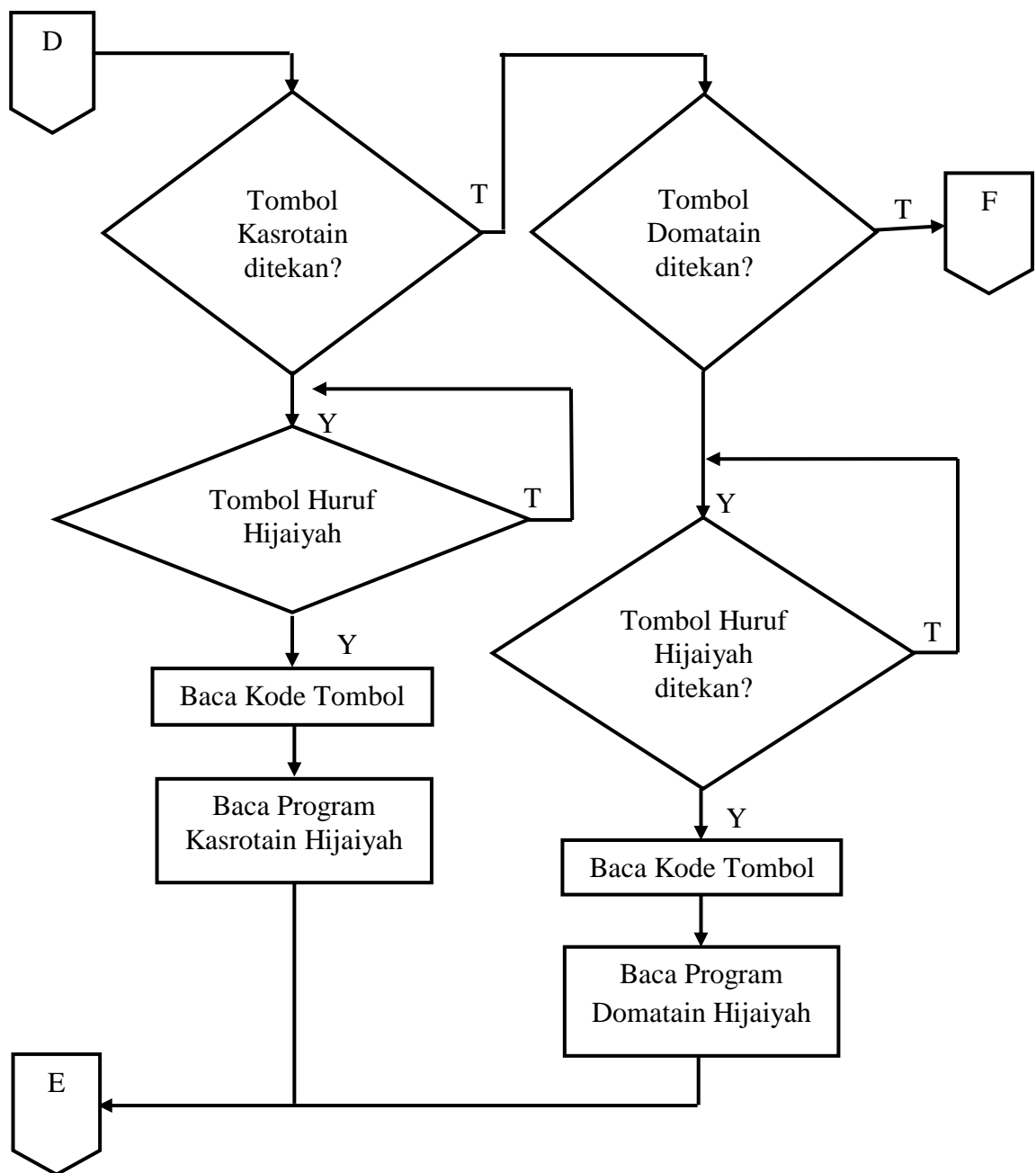
- Step 9* : Apakah tombol huruf Kasroh ditekan? Jika Ya maka lanjut ke Step 10, Jika tidak maka lanjut ke step 13.
- Step 10* : Apakah tombol huruf hijaiyah ditekan? Jika Ya maka lanjut step 11. Jika tidak kembali ke step sebelumnya.
- Step 11* : Baca Kode Tombol.
- Step 12* : Output LCD menampilkan bacaan huruf dan *Speaker* akan mengeluarkan suara dari huruf hijaiyah Kasroh.
- Step 13* : Apakah tombol huruf Domah ditekan? Jika Ya maka lanjut ke Step 14, Jika tidak maka lanjut ke step 17.
- Step 14* : Apakah tombol huruf hijaiyah ditekan? Jika Ya maka lanjut step 15. Jika tidak kembali ke step sebelumnya
- Step 15* : Baca Kode Tombol.
- Step 16* : Output LCD menampilkan bacaan huruf dan *Speaker* akan mengeluarkan suara dari huruf hijaiyah Domah.
- Step 17* : Apakah tombol huruf Fathatain ditekan? Jika Ya maka lanjut ke Step 18, Jika tidak maka lanjut ke step 21.
- Step 18* : Apakah tombol huruf hijaiyah ditekan? Jika Ya maka lanjut step 19. Jika tidak kembali ke step sebelumnya.
- Step 19* : Baca Kode Tombol.
- Step 20* : Output LCD menampilkan bacaan huruf dan *Speaker* akan mengeluarkan suara dari huruf hijaiyah Fathatain.
- Step 21* : Apakah tombol huruf Kasrotain ditekan? Jika Ya maka lanjut ke Step 22, Jika tidak maka lanjut ke step 25.

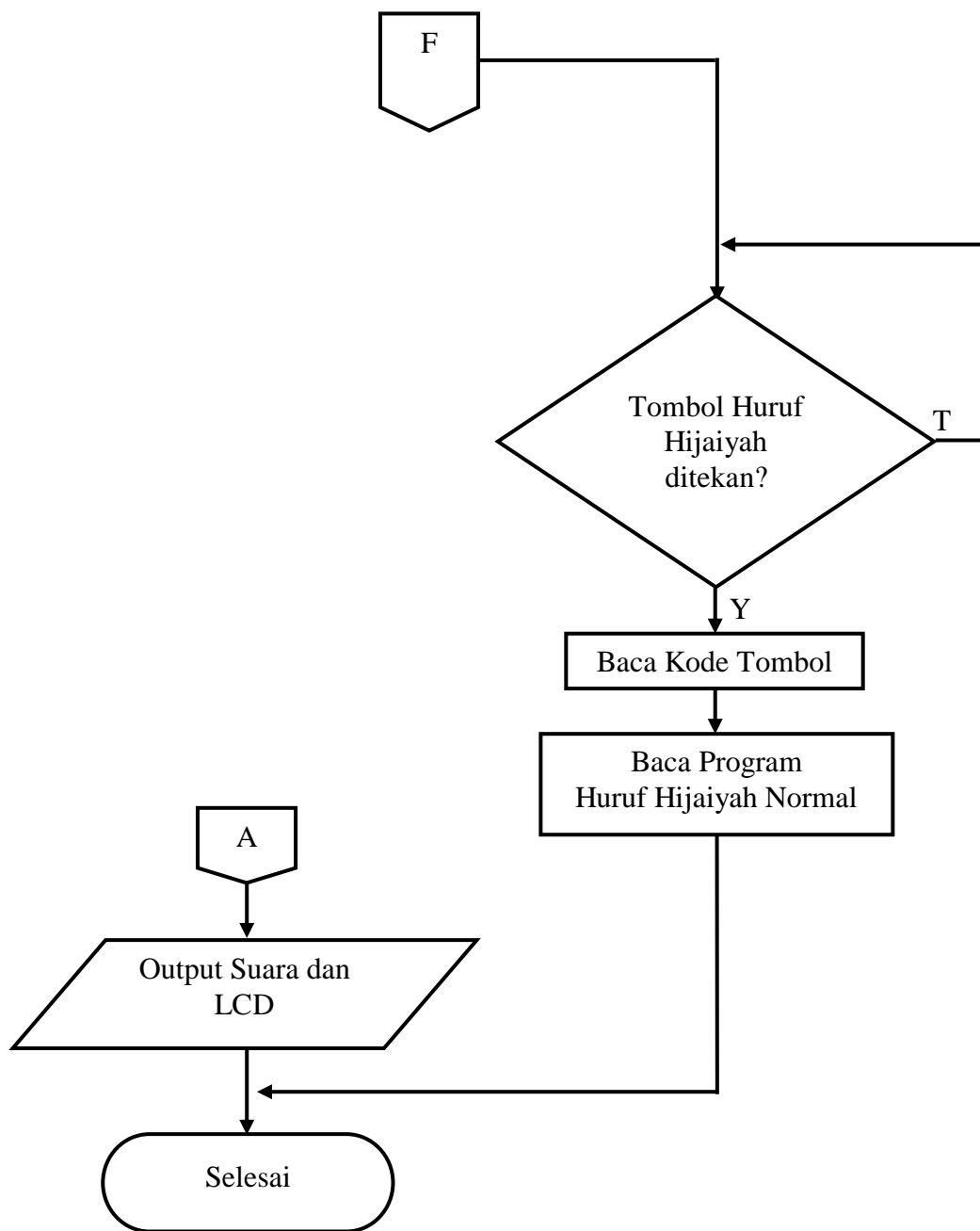
- Step 22* : Apakah tombol huruf hijaiyah ditekan? Jika Ya maka lanjut step 23, Jika tidak kembali ke step sebelumnya.
- Step 23* : Baca Kode Tombol.
- Step 24* : Output LCD menampilkan bacaan huruf dan *Speaker* akan mengeluarkan suara dari huruf hijaiyah Kasrotain.
- Step 25* : Apakah tombol huruf Domatain ditekan? Jika Ya maka lanjut ke Step 26, Jika tidak maka lanjut ke step 29.
- Step 26* : Apakah tombol huruf hijaiyah ditekan? Jika Ya maka lanjut step 27, Jika tidak kembali ke step sebelumnya.
- Step 27* : Baca Kode Tombol.
- Step 28* : Output LCD menampilkan bacaan huruf dan *Speaker* akan mengeluarkan suara dari huruf hijaiyah Domatain.
- Step 29* : Apakah tombol huruf hijaiyah ditekan? Jika Ya maka lanjut step 30. Jika tidak kembali ke step sebelumnya.
- Step 30* : Baca Kode Tombol.
- Step 31* : Output LCD menampilkan bacaan huruf dan *Speaker* akan mengeluarkan suara dari huruf hijaiyah Normal.
- Step 32* : Selesai.

b. Diagram Alir/Flowchart Program









Gambar 18. *Flowchart*

D. Pembuatan Alat

Dalam pembuatan media pembelajaran huruf hijaiyah ini terdapat beberapa hal yang harus diperhatikan dalam pembuatan alat terlebih dahulu menyiapkan alat, bahan, kemudian pembuatan PCB, pemasangan komponen, pembuatan box dan perakitan.

1. Alat dan Bahan

Alat dan bahan adalah bagian pokok yang paling penting di dalam pembuatan media pembelajaran huruf hijaiyah ini. Berikut adalah Tabel alat dan bahan yang dibutuhkan antara lain :

Tabel 9. Alat dan Bahan

No.	Alat dan Bahan
1.	Unit Komputer / Laptop
2.	<i>USB Downloader</i>
3.	<i>Ferichloride</i> (FeCl_3)
4.	Bor PCB
5.	Multimeter
6.	Tang
7.	Solder
8.	Timah
9.	Setrika
10.	PCB polos
11.	Komponen meliputi ATMega16, <i>Crystal</i> , IC penguat OP-AMP, dan komponen pendukung lainnya.
12.	LCD (<i>Liquid Crystal Display</i>) ukuran 16 x 2

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Alat dan Bahan
13.	Box
14.	Lem
15.	Mur baut
16.	Penggaris
17.	<i>Acrylic</i>
18.	Atraktor
19.	Kabel

2. Pembuatan PCB

a. Pembuatan *Layout* PCB

Dalam pembuatan *layout* PCB dapat dibuat dengan menggunakan *software Proteus Professional 7.10 SPO*. Sebelum membuat rangkaian hal utama yang dilakukan yaitu mencari komponen - komponen yang dibutuhkan seperti IC ATmega16, IC Regulator 7805, *Crystal*, resistor, kapasitor, diode, dan lain sebagainya. Setelah itu mebuat rangkaian sistem minimum ATmega16, rangkaian OP-AMP, rangkaian catu daya, rangkaian penampil Display LCD.

b. Penyablonan PCB

Proses penyablonan dilakukan dengan cara :

- 1) menyiapkan strika menunggu hingga panas, menyiapkan kertas sablonan yang berisi rangkaian, PCB kosong.
- 2) Kemudian haluskan terlebih dahulu PCB dengan rempelas agar tinta dapat menempel dengan baik.

- 3) Kemudian bersihkan PCB dari bekasnya, lalu tempelkan kertas rangkaian ke PCB dan gosok secara merata sampai tinta dalam kertas sablon benar - benar sudah menempel di PCB.
- 4) Lalu bilas PCB dengan air sambil lepas kertas sablon secara perlahan.

c. Pelarutan dan Pengeboran PCB

Proses pelarutan dilakukan dengan cara :

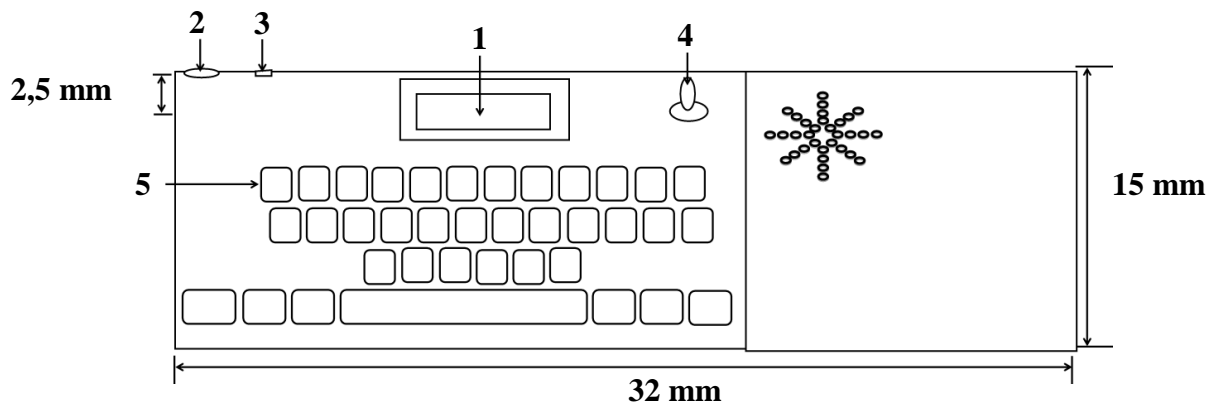
- 1) menyiapkan tempat yang akan digunakan untuk tempat air dan larutan *Ferichloride*.
- 2) Mengaduk air dengan larutan *Ferichloride* secara merata sampai air berubah warna.
- 3) Setelah itu masukan PCB ke dalam larutan *Ferichloride*, tunggu sampai tembaga pada PCB terkelupas dan rangkaian dapat dilihat secara jelas.
- 4) Rangkaian PCB yang sudah jadi dapat dilubangi dengan menggunakan bor PCB.
- 5) Lubang pada PCB akan dipakain sebagai tempat untuk pemasangan kaki-kaki komponen.

3. Pemasangan Komponen

Berikut adalah langkah - langkah dalam pemasangan komponen secara keseluruhan antara lain :

- a. Menyiapkan komponen - komponen yang akan digunakan.

- b. Memasang komponen yang paling terkecil terlebih dahulu kemudian komponen yang paling besar.
 - c. Setelah terpasang, komponen disolder dengan menggunakan timah.
 - d. Pemotongan kaki komponen secara rapih.
4. Pembuatan Rangkaian Box
 - a. Perancangan Ukuran Box



Gambar 19. Desain Rangkaian Box

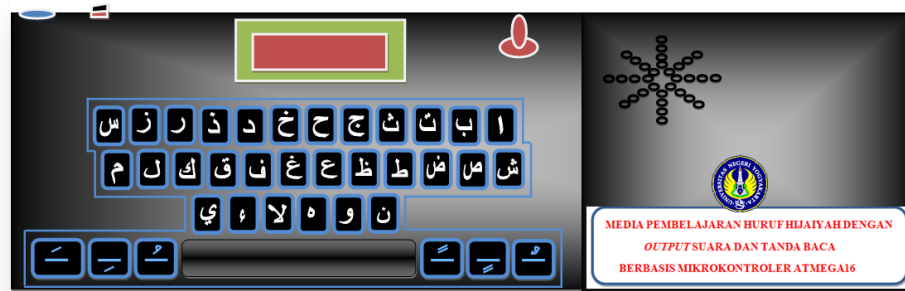
Keterangan *Box* :

- 1) LCD (*Liquid Crystall Display*)
- 2) Soket *Power Supply*
- 3) Saklar *Power*
- 4) Pengatur volume suara
- 5) Tombol *Keyboard*

- b. Pembuatan Rangkaian Box

Di dalam pembuatan rangkaian box, bahan yang dipakai menggunakan bahan *acrylic* dengan ketebalan 0.3 cm. Pembuatan box ini akan digunakan untuk menutupi tombol huruf ataupun angka pada

keyboard yang sudah tidak terpakai dan berfungsi sebagai tempat untuk menutupi rangkaian PCB sistem minimum, kabel dan *speaker*. Di bawah ini adalah Gambar rangkaian box yang sudah jadi.



Gambar 20. Rangkaian Box

E. Pengujian Alat

Pengujian alat dilakukan dengan cara mengambil data dan hasil pengukuran, pengujian didapatkan dari hasil pengujian Uji Fungsional Setiap Bagian dan Pengujian Keluruhan Uji. Berikut ini adalah beberapa langkah – langkah pengujian yang dilakukan :

1. Pengujian Uji Fungsional Setiap Bagian

Penujian dilakukan untuk mengetahui apakah setiap perangkat dapat fungsi dengan baik sesuai keinginan. Berikut ini adalah beberapa pengujian yang dilakukan.

- a. Tegangan Catu Daya, tegangan ATmega 16, tegangan *SD Card*, rangkaian TDA 7052.

Tabel 10. Pengujian tegangan Catu Daya, tegangan ATmega 16, tegangan *SD Card*, tegangan TDA 7052.

No.	Bagian Pengukuran	Hasil Pengukuran
1.	• Tegangan Catu Daya	
2.	• Tegangan VCC IC ATmega16	
3.	• Tegangan <i>Supply</i> IC TDA7052	
4.	<ul style="list-style-type: none"> • Tegangan <i>SD Card</i> Pin 1 Pin 2 Pin 3 Pin 4 Pin 5 Pin 6 Pin 7 Pin 8 Pin 9 	

b. Pengujian Kode Tombol Keyboard

Tabel 11. Pengujian Kode Tombol Keyboard

No.	Tombol Huruf Keyboard	Kode Keyboard	Tampilan LCD
1.]		
2.	[
3.	p		
4.	o		
5.	i		
6.	u		
7.	y		

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Tombol Huruf Keyboard	Kode Keyboard	Tampilan LCD
8.	t		
9.	r		
10.	e		
11.	w		
12.	q		
13.	\		
14.	‘		
15.	;		
16.	l		
17.	k		
18.	j		
19.	h		
20.	g		
21.	f		
22.	d		
23.	s		
24.	a		
25.	,		
26.	m		
27.	n		
28.	b		
29.	v		
30.	c		

c. Pengujian Sistem Minimum ATmega16

- 1) Pengujian Tampilan LCD
- 2) Pengujian Tombol *Harokat*
- 3) Pengujian Tampilan Suara

2. Pengujian Keseluruhan Uji Sistem

Pada pengujian keseluruhan uji sistem ini dilakukan untuk mengetahui kinerja dari keseluruhan alat dan untuk mengetahui kesesuaian antara *output* suara huruf hijaiyah dengan *input* tombol huruf hijaiyah. Pengujian ini dimaksudkan untuk mengetahui validasi *output* suara dan tampilan cara baca huruf hijaiyah pada LCD. Berikut adalah bagian yang diujikan yakni :

a. Pengujian Tombol Huruf Hijaiyah

Tabel 12. Pengujian Tombol Huruf Hijaiyah

No.	Huruf Hijaiyah	Tombol Huruf Keyboard
1.	ا	
2.	ب	
3.	ت	
4.	ث	
5.	ج	
6.	ح	
7.	خ	
8.	د	
9.	ذ	
10.	ر	
11.	ز	
12.	س	
13.	ش	
14.	ص	
15.	ض	
16.	ط	
17.	ظ	

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Huruf Hijaiyah	Tombol Huruf Keyboard
18.	ع	
19.	غ	
20.	ف	
21.	ق	
22.	ك	
23.	ل	
24.	م	
25.	ن	
26.	و	
27.	ه	
28.	لا	
29.	ء	
30.	ي	

b. Pengujian Tombol Huruf Hijaiyah Normal

Tabel 13. Pengujian Tombol Huruf Hijaiyah Normal

No.	Huruf Hijaiyah	Tampilan LCD	Suara
1.	ا		
2.	ب		
3.	ت		
4.	ث		
5.	ج		
6.	ح		
7.	خ		

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Huruf Hijaiyah	Tampilan LCD	Suara
8.	د		
9.	ذ		
10.	ر		
11.	ز		
12.	س		
13.	ش		
14.	ص		
15.	ض		
16.	ط		
17.	ظ		
18.	ع		
19.	غ		
20.	ف		
21.	ق		
22.	ك		
23.	ل		
24.	م		
25.	ن		
26.	و		
27.	ه		
28.	لا		
29.	ء		
30.	ي		

- c. Pengujian Kombinasi Tombol Huruf Hijaiyah dan *Harokat* yang terdiri dari (*Fathah, Kasroh, Domah, Fathatain, Kasrotain, Domatain*).

Tabel 14. Pengujian Kombinasi Tombol Huruf Hijaiyah dan *Harokat* yang terdiri dari (*Fathah, Kasroh, Domah, Fathatain, Kasrotain, Domatain*).

No.	Huruf Hijaiyah	<i>Harokat</i> Huruf	Tampilan LCD	Suara
1.	أَ			
2.	بَ			
3.	تَ			
4.	ثَ			
5.	جَ			
6.	حَ			
7.	خَ			
8.	دَ			
9.	ذَ			
10.	رَ			
11.	زَ			
12.	سَ			
13.	شَ			
14.	صَ			
15.	ضَ			
16.	طَ			
17.	ظَ			
18.	عَ			
19.	غَ			
20.	فَ			
21.	قَ			
22.	كَ			
23.	لَ			
24.	مَ			
25.	نَ			

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Huruf Hijaiyah	<i>Harokat</i> Huruf	Tampilan LCD	Suara
26.	و			
27.	ة			
28.	لا			
29.	ء			
30.	ي			

d. Pengukuran Kecepatan Tombol *Input* dan *Output* Suara

Tabel 15. Pengukuran Kecepatan Tombol *Input* dan *Output* Suara

No.	Huruf Hijaiyah	Kecepatan <i>Output</i> Suara/(detik)
1.	ا	
2.	ب	
3.	ت	
4.	ث	
5.	ج	
6.	ح	
7.	خ	
8.	د	
9.	ذ	
10.	ر	
11.	ز	
12.	س	
13.	ش	
14.	ص	
15.	ض	
16.	ط	
17.	ظ	

Lanjut ke tabel berikutnya :

Sambungan ke table sebelumnya :

No.	Huruf Hijaiyah	Kecepatan <i>Output</i> Suara/(detik)
18.	ع	
19.	غ	
20.	ف	
21.	ق	
22.	ك	
23.	ل	
24.	م	
25.	ن	
26.	و	
27.	ه	
28.	لا	
29.	ء	
30.	ي	

F. Jadwal Pelaksanaan

Jadwal pelaksanaan pembuatan proyek akhir ini dimulai pada bulan Juni 2013 sampai Mei 2014 dengan rincian pelaksanaan kegiatan seperti pada Tabel berikut.

Tabel 16. Jadwal Pelaksanaan

No.	Kegiatan	Bulan Ke – , Tahun 2013/2014									
		II	III	IV	V	VI	VII	VIII	IX	X	
1.	Penelusuran literatur										
2.	Penyusunan proposal										
3.	Tahap perencanaan										
4.	Proses pembuatan dan bimbingan										
5.	Tahap pengujian										
6.	Evaluasi dan perbaikan										
7.	Penyusunan laporan										

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Kegiatan	Bulan Ke – , Tahun 2013/2014									
		II	III	IV	V	VI	VII	VIII	IX	X	
8.	Ujian proyek akhir										
9.	Revisi dan Penyerahan Laporan										

G. Daftar Komponen dan Biaya Pembuatan

Di bawah ini adalah tabel daftar pembelian komponen - komponen beserta anggaran biaya yang digunakan untuk membuat “Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16.”

Tabel 17. Daftar Komponen dan Biaya Pembuatan

No.	Keterangan	Jumlah	Satuan	Harga	
				Satuan(Rp)	Jumlah(Rp)
1.	Modul WT5001M02 – 28P	1	Buah	120.000	120.000
2.	IC TDA7052	1	Buah	5.000	5.000
3.	IC ATmega16	1	Buah	30.000	30.000
4.	LCD 16 x 2	1	Buah	40.000	40.000
5.	Speaker 2,5watt	1	Buah	15.000	15.000
6.	7805	1	Buah	2.000	2.000
7.	Pembuatan Box	1	Buah	85.000	85.000
8.	Keyboard	1	Buah	35.000	35.000
9.	Jack Kontra DC	1	Buah	1.500	1.500
10.	Switch Push Button	6	Buah	500	3.000
11.	Trimpot	1	Buah	1.000	1.000
12.	Resistor	2	Buah	100	200
13.	Crystal	1	Buah	2.000	2.000
14.	Capasitor keramik	4	Buah	100	400
15.	Kabel	2	Meter	10.000	20.000

lanjut ke tabel berikutnya :

Sambungan ke tabel sebelumnya :

No.	Keterangan	Jumlah	Satuan	Harga	
				Satuan(Rp)	Jumlah(Rp)
16.	Stiker Huruf Hiajaiyah	1	Lembar	3.000	3.000
17.	Glossy Rangakaian	2	Lembar	2.000	4.000
18.	Soket Power Supply	1	Buah	2.000	2.000
19.	Soket Sisir Male/ Female	4	Buah	2.000	8.000
20.	Resistor Variable	1	Buah	5.000	5.000
Jumlah Total (Rp.)					370.100

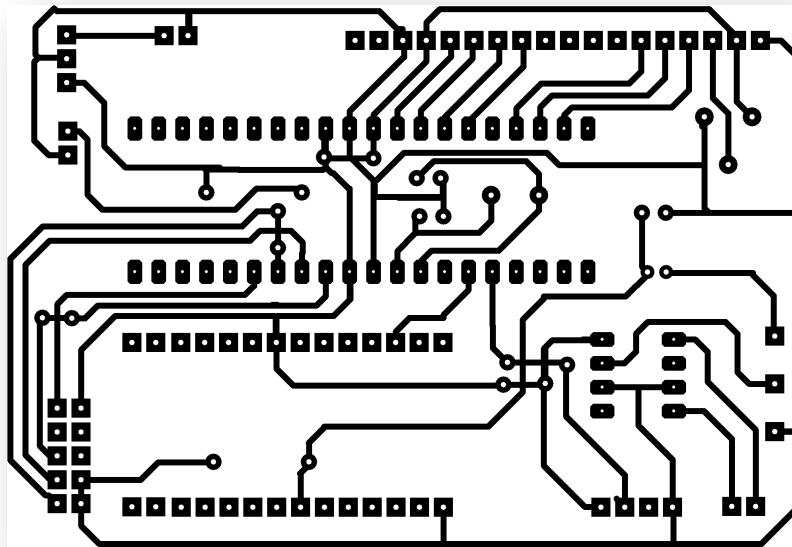
BAB IV

HASIL, PENGUJIAN DAN PEMBAHASAN

A. Hasil

Hasil perancangan Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16 ini terdiri dari hasil *layout* PCB, hasil perakitan alat dan hasil perakitan alat jadi seperti berikut ini.

1. Hasil *Layout* PCB Rangkaian Sistem Minimum ATmega16, Rangkaian Penguat OP-AMP, Rangkaian Display LCD.



Gambar 21. Rangkaian Sistem Minimum ATmega16, Rangkaian Penguat OP-AMP, Rangkaian Display LCD.

Rangkaian Sistem Minimum didesain secara sistematis agar tidak memakan tempat pada box rangkaian. Rangkaian ini dibuat dengan menggunakan *software proteus professional*.

2. Hasil Perancangan Alat

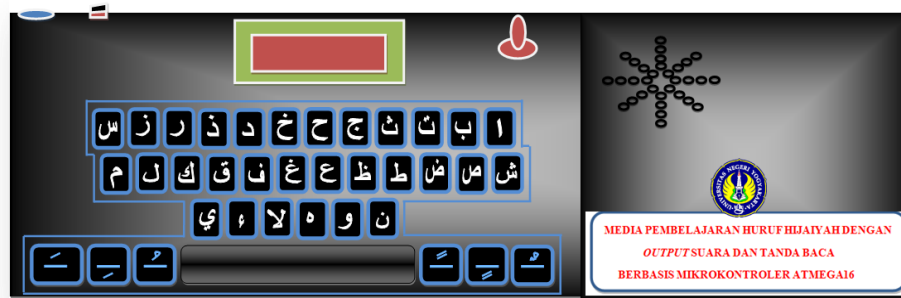
Berikut adalah gambar hasil perancangan alat yang dibuat sesederhana mungkin, mengikuti alur sketsa dari rancangan yang telah dibuat. Rancangan membutuhkan tempat untuk menaruh sistem minimum dan *speaker* serta tutup box sebagai penutupnya.



Gambar 22. Hasil perancangan Alat

3. Hasil Perancangan Alat Jadi

Hasil dari perancangan alat jadi ini dibutuhkan kejelian dalam menempatkan tombol – tombol mana saja yang sesuai, sehingga dapat digunakan sebagai tempat untuk tombol huruf dan tanda baca. Dari hasil ini membutuhkan desain warna yang sesuai dengan kegunaan sebagai media pembelajaran huruf hijaiyah untuk anak – anak. Berikut adalah gambar hasil perancangan alat yang sudah jadi dan siap untuk digunakan.



Gambar 23. Hasil Perancangan Alat Jadi

B. Pengujian

Pengujian media pembelajaran huruf hijaiyah terdiri dari 3 tahap pengujian yakni pengukuran tegangan kerja *output* huruf hijaiyah, pengujian tombol huruf hijaiyah, pengukuran kecepatan tombol *input* dan *output* suara pengukuran kecepatan tombol *input* dan *output* suara.

1. Pengujian Uji Fungsional Setiap Bagian

- a. Pengujian Tegangan Catu Daya, tegangan ATmega 16, tegangan *SD Card*, tegangan TDA 7052.

Pengukuran tegangan kerja dilakukan pada bagian tertentu dari tiap-tiap rangkaian. Pengukuran dengan menggunakan alat ukur multimeter dan *power supply variable*. Tabel 18 merupakan data hasil pengukuran dari tegangan tiap-tiap rangkaian.

Tabel 18. Data Pengukuran tegangan

No.	Bagian Pengukuran	Hasil Pengukuran
1.	• Tegangan Catu Daya	12 V
2.	• Tegangan VCC IC ATmega16	5 V

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Bagian Pengukuran	Hasil Pengukuran
3.	• Tegangan <i>Supply</i> IC TDA7052	5 V
4.	• Tegangan <i>SD Card</i>	
	Pin 1	3.2 V
	Pin 2	0 V
	Pin 3	3.2 V
	Pin 4	3.2 V
	Pin 5	3.2 V
	Pin 6	0 V
	Pin 7	3.4 V
	Pin 8	1.8 V
	Pin 9	2.5 V

b. Pengujian Tombol Kode Keyboard

Pada program ini difungsikan untuk membaca data bit yang kesemuanya terdiri dari 11 bit, namun data yang dipakai hanya 8 bit dan mengabaikan start bit, top bit dan parity bit. Jika data lebih dari 2 dan kurang dari 11 maka data bit disimpan pada variable data dengan nama variable “data”. Data bit itu akan diolah menjadi heksa decimal dengan cara penjumlahan. Hasil dari perhitungan itu digunakan untuk mencari kode keyboard yang akan di eksekusi. Berikut ini adalah program untuk mencari kode keyboard yang akan digunakan dalam mengeksekusi program.

```
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here

if (count > 2 && count <11)
```







```

// Read the data bits,neglecting the start bit and
the stop and parity bits
{
data = (data >> 1);
if(PIND & 8)
{data = data | 0x80;x=1;} // Store a '1'
}
if(--count ==0 ) // All bits received
{count = 11;
PORTA=data;
data=0;}
}

```


Pada pengujian tombol kode keyboard ini dilakukan untuk mengetahui masing – masing data kode dari tiap tombol keyboard. Berikut ini adalah tabel kode keyboard yang disebagai kode perintah untuk mengeksekusi program.

Tabel 19. Hasil Pengujian Tombol Kode Keyboard

No.	Tombol Huruf Keyboard	Kode Keyboard	Tampilan LCD
1.]	91	
2.	[84	
3.	P	77	
4.	O	68	
5.	I	67	
6.	U	60	







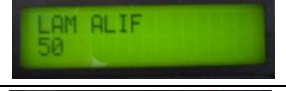


Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Tombol Huruf Keyboard	Kode Keyboard	Tampilan LCD
7.	Y	53	
8.	T	44	
9.	R	45	
10.	E	36	
11.	W	29	
12.	Q	21	
13.	\	93	
14.	‘	82	
15.	;	76	
16.	L	75	
17.	K	66	
18.	J	59	
19.	H	51	
20.	G	52	
21.	F	43	

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Tombol Huruf Keyboard	Kode Keyboard	Tampilan LCD
22.	D	35	
23.	S	27	
24.	A	28	
25.	,	65	
26.	M	58	
27.	N	49	
28.	B	50	
29.	V	42	
30.	C	33	

c. Pengujian Sistem Minimum ATmega16

1) Pengujian Tampilan LCD

Tampilan awal pada LCD didahului oleh munculnya karakter huruf yang sebelumnya telah dimasukan program. Di bawah ini adalah program yang telah di-*download* pada mikrokontroler ATmega16 sebagai berikut :

```

lcd_init(16);
lcd_clear();
a=0;
lcd_gotoxy(0,0);
lcd_putsf(" Loading...");

```

```
delay_ms(4500);  
lcd_clear();
```

Pada pengujian huruf ini untuk mengetahui kesesuaian antara suara huruf hijaiyah dengan tombol huruf hijaiyah. Pengujian juga dilakukan untuk mengetahui validasi *output* suara dan tampilan cara baca huruf hijaiyah pada LCD. Berikut adalah Gambar tampilan kinerja LCD dalam menampilkan karakter sesuai dengan program yang di-*download* pada mikrokontroler ATmega16.



Gambar 24. Tampilan Awal LCD

2) Pengujian Tombol *Harokat*

Setiap masukan dari sebuah tombol akan menghasilkan keluaran berupa hasil eksekusi dari perintah masukan. Pada *Input*-an tombol *harokat* ini digunakan sebagai perintah untuk membedakan antara huruf hijaiyah satu dengan yang lainnya. Berikut adalah program tombol *harokat* yang telah di-*download* pada mikrokontroler ATmega16 sebagai berikut :

```
lcd_clear();  
a=PORTA/2;  
lcd_gotoxy(0,0);  
lcd_putsf("Baca :");  
if (PINA.0==1)  
{ fathah(); }  
else if (PINA.1==1)  
{ kasroh(); }  
else if (PINA.2==1)
```



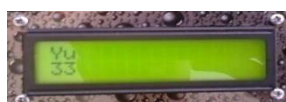



```

{dhomah();}
else if (PINA.4==1)
{fathah_tain();}
else if (PINA.5==1)
{ kasroh_tain();}
else if (PINA.6==1)
{ dhomah_tain();}
else if (PINA.0==0 && PINA.1==0 && PINA.2==0 &&
PINA.3==0 && PINA.4==0)
{hijaiyah() }
delay_ms(300);

```

Pada pengujian tombol *harokat* ini dimaksudkan untuk mengetahui keseuaian tanda baca huruf yang telah dikombinasikan dengan ke-30 huruf hijaiyah. berikut adalah sampel dari pengkombinasian tanda baca huruf (*harokat*) dengan ke-30 huruf hijaiyah :

Tabel 20. Hasil Pengujian Tombol *Harokat*

Huruf Hijaiyah	<i>Harokat</i> Huruf	Tampilan LCD
ﻯ	<i>Fathah</i>	
	<i>Kasroh</i>	
	<i>Domah</i>	
	<i>Fathatain</i>	
	<i>Kasrotain</i>	
	<i>Domatain</i>	

3) Pengujian Tampilan Suara

Suara yang dihasilkan berasal dari modul MP3 Player yang digunakan untuk menjalankan *file* suara yang tersimpan dalam *Micro Sd*. Bentuk *file* yang berada di dalam *Micro Sd* disusun secara berurutan sesuai dengan urutan huruf. Berikut ini adalah program untuk menampilkan suara.

```
void play(int number)
{  putchar(0x7E);
  putchar(0x04);
  putchar(0xA0);
  putchar(number>>8);
  putchar(number&0xff);
  putchar(0x7E);
  delay_ms(100); }
void stop()
{  putchar(0x7e);
  putchar(0x02);
  putchar(0xA4);
  putchar(0x7e); }
void suara(int no)
{if(x==1 && a>0)
{play(no);
x=2;
delay_ms(20);}
}
void hijaiyah()
{is_up =0;
{switch (a)
```

pada pengujian tampilan suara ini bertujuan untuk mengetahui keluaran yang dihasilkan dari hasil masukan berupa perintah tekan tombol huruf. Setiap tombol huruf yang ditekan memiliki durasi waktu yang berbeda – beda. Hal ini disebabkan karena pengaturan delay pada program secara manual sesuai dengan kebutuhan. Berikut ini adalah tabel tampilan suara.

Tabel 21. Hasil Pengujian Tampilan Suara

No.	Huruf Hijaiyah	<i>Input</i> Keyboard/(detik)	<i>Output</i> Suara
1.	ا	1 detik	ada
2.	ب	1 detik	ada
3.	ت	2 detik	ada

2. Pengujian Keseluruhan Uji Sistem

a. Pengujian Tombol Huruf Hijaiyah

Pengujian tombol huruf hijaiyah ini dimaksudkan untuk menyesuaikan antara tombol keyboard yang digunakan sebagai tombol huruf hijaiyah. Hal ini dilakukan agar tidak terjadi kekeliruan dalam memanggil kode keyboard. Berikut adalah table pengujian tombol huruf hijaiyah :

Tabel 22. Hasil Pengujian Tombol Huruf Hijaiyah

No.	Huruf Hijaiyah	Tombol Huruf Keyboard
1.	ا]
2.	ب	[
3.	ت	p
4.	ث	o
5.	ج	i
6.	ح	u
7.	خ	y
8.	د	t
9.	ذ	r
10.	ر	e

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :






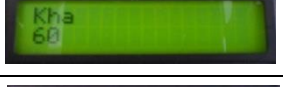


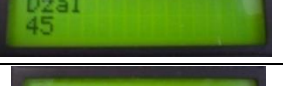




No.	Huruf Hijaiyah	Tombol Huruf Keyboard
11.	ز	w
12.	س	q
13.	ش	\
14.	ص	‘
15.	ض	;
16.	ط	l
17.	ظ	k
18.	ع	j
19.	غ	h
20.	ف	g
21.	ق	f
22.	ك	d
23.	ل	s
24.	م	a
25.	ن	,
26.	و	m
27.	ه	n
28.	لا	b
29.	ء	v
30.	ي	c

b. Pengujian Tombol Huruf Hijaiyah Normal

Pengujian perhitungan ini bertujuan untuk mengetahui hasil data kebenaran dari *output* huruf hijaiyah. Pengujian dilakukan dengan mengoperasikan tombol huruf pada setiap fungsi yang ada. Setiap fungsi diujicoba dengan menggunakan beberapa sampel perhitungan /

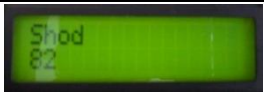
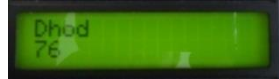
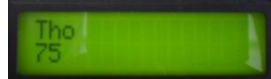












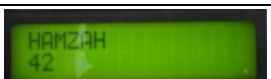

pengujian. Berikut tabel data hasil pengujian berdasarkan masing-masing fungsi yang ada.

Tabel 23. Hasil Pengujian Tombol Huruf Hijaiyah Normal

No.	Huruf Hijaiyah	Tampilan LCD	Suara
1.	ا		Ada
2.	ب		Ada
3.	ت		Ada
4.	ث		Ada
5.	ج		Ada
6.	ح		Ada
7.	خ		Ada
8.	د		Ada
9.	ذ		Ada
10.	ر		Ada
11.	ز		Ada
12.	س		Ada
13.	ش		Ada

Lanjut ke tabel berikutnya :








Sambungan tabel sebelumnya :

14.	ص		Ada
15.	ض		Ada
16.	ط		Ada
17.	ظ		Ada
18.	ع		Ada
19.	غ		Ada
20.	ف		Ada
21.	ق		Ada
22.	ك		Ada
23.	ل		Ada
24.	م		Ada
25.	ن		Ada
26.	و		Ada
27.	ه		Ada
28.	لا		Ada
29.	ء		Ada
30.	ي		Ada

Tabel di atas merupakan data hasil pengujian ke-30 tombol huruf hijaiyah. Pengujian dilakukan dengan cara menekan tombol huruf pada *keyboard*. Setiap tombol yang ditekan akan menampilkan hasil *output* yang berbeda - beda. Tampilan *output* pada LCD yang merepresentasikan data hasil pengujian menunjukkan kesesuaian dengan tombol yang ditekan.


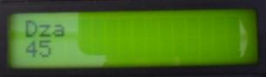













- c. Pengujian Kombinasi Tombol huruf Hijaiyah dan *Harokat* (*Fathah, Kasroh, Domah, Fathatain, Kasrotain, Domatain*)

Tabel 24. Data Pengujian Kombinasi Huruf dan Tanda Baca Fathah

No.	Huruf Hijaiyah	Tanda Baca Huruf	Tampilan LCD	Suara
1.	أ	Fathah		Ada
2.	ب	Fathah		Ada
3.	ت	Fathah		Ada
4.	ث	Fathah		Ada
5.	ج	Fathah		Ada
6.	ح	Fathah		Ada
7.	خ	Fathah		Ada


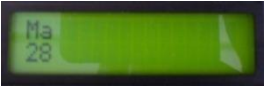
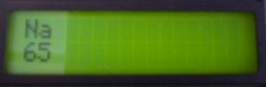





Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :





No.	Huruf Hijaiyah	Tanda Baca Huruf	Tampilan LCD	Suara
8.	دَ	Fathah		Ada
9.	ذَ	Fathah		Ada
10.	رَ	Fathah		Ada
11.	زَ	Fathah		Ada
12.	سَ	Fathah		Ada
13.	شَ	Fathah		Ada
14.	صَ	Fathah		Ada
	ضَ	Fathah		Ada
16.	طَ	Fathah		Ada
17.	ظَ	Fathah		Ada
18.	عَ	Fathah		Ada
19.	غَ	Fathah		Ada
20.	فَ	Fathah		Ada
21.	قَ	Fathah		Ada
22.	كَ	Fathah		Ada

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :









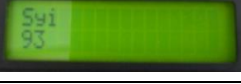
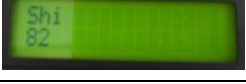

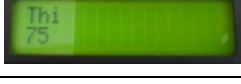


No.	Huruf Hijaiyah	Tanda Baca Huruf	Tampilan LCD	Suara
23.	ل	Fathah		Ada
24.	م	Fathah		Ada
25.	ن	Fathah		Ada
26.	و	Fathah		Ada
27.	ه	Fathah		Ada
28.	لا	Fathah		Ada
29.	ء	Fathah		Ada
30.	ي	Fathah		Ada

Tabel 25. Data Pengujian Kombinasi Huruf dan Tanda Baca Kasroh

No.	Huruf Hijaiyah	Tanda Baca Huruf	Tampilan LCD	Suara
1.	ا	Kasroh		Ada
2.	ب	Kasroh		Ada
3.	ت	Kasroh		Ada
4.	ث	Kasroh		Ada






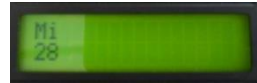






Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :







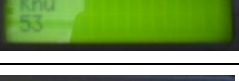





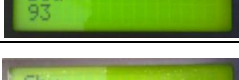
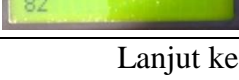
No.	Huruf Hijaiyah	Tanda Baca Huruf	Tampilan LCD	Suara
5.	ج	Kasroh		Ada
6.	ح	Kasroh		Ada
7.	خ	Kasroh		Ada
8.	د	Kasroh		Ada
9.	ذ	Kasroh		Ada
10.	ر	Kasroh		Ada
11.	ز	Kasroh		Ada
12.	س	Kasroh		Ada
13.	ش	Kasroh		Ada
14.	ص	Kasroh		Ada
15.	ض	Kasroh		Ada
16.	ط	Kasroh		Ada
17.	ظ	Kasroh		Ada
18.	ع	Kasroh		Ada

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

















No.	Huruf Hijaiyah	Tanda Baca Huruf	Tampilan LCD	Suara
19.	غ	Kasroh		Ada
20.	ف	Kasroh		Ada
21.	ق	Kasroh		Ada
22.	ك	Kasroh		Ada
23.	ل	Kasroh		Ada
24.	م	Kasroh		Ada
25.	ن	Kasroh		Ada
26.	و	Kasroh		Ada
27.	ه	Kasroh		Ada
28.	ي	Kasroh		Ada
29.	ء	Kasroh		Ada
30.	ي	Kasroh		Ada

Tabel 26. Data Pengujian Kombinasi Huruf dan Tanda Baca Domah














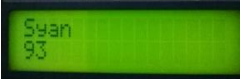

No.	Huruf Hijaiyah	Tanda Baca Huruf	Tampilan LCD	Suara
1.	أ	Domah		Ada
2.	ب	Domah		Ada
3.	ت	Domah		Ada
4.	ث	Domah		Ada
5.	ج	Domah		Ada
6.	ح	Domah		Ada
7.	خ	Domah		Ada
8.	د	Domah		Ada
9.	ذ	Domah		Ada
10.	ر	Domah		Ada
11.	ز	Domah		Ada
12.	س	Domah		Ada
13.	ش	Domah		Ada
14.	ص	Domah		Ada

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :
















No.	Huruf Hijaiyah	Tanda Baca Huruf	Tampilan LCD	Suara
15.	ضُ	Domah		Ada
16.	طُ	Domah		Ada
17.	ظُ	Domah		Ada
18.	عُ	Domah		Ada
19.	غُ	Domah		Ada
20.	فُ	Domah		Ada
21.	قُ	Domah		Ada
22.	كُ	Domah		Ada
23.	لُ	Domah		Ada
24.	مُ	Domah		Ada
25.	نُ	Domah		Ada
26.	وُ	Domah		Ada
27.	هُ	Domah		Ada
28.	لَا	Domah		Ada
29.	ءُ	Domah		Ada
30.	يُ	Domah		Ada

Tabel 27. Data Pengujian Kombinasi Huruf dan Tanda Baca Fathatain


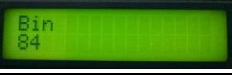











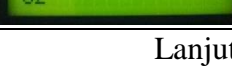
No.	Huruf Hijaiyah	Tanda Baca Huruf	Tampilan LCD	Suara
1.	أ	Fathatain		Ada
2.	ب	Fathatain		Ada
3.	ت	Fathatain		Ada
4.	ث	Fathatain		Ada
5.	ج	Fathatain		Ada
6.	ح	Fathatain		Ada
7.	خ	Fathatain		Ada
8.	د	Fathatain		Ada
9.	ذ	Fathatain		Ada
10.	ر	Fathatain		Ada
11.	ز	Fathatain		Ada
12.	س	Fathatain		Ada
13.	ش	Fathatain		Ada
14.	ص	Fathatain		Ada
15.	ض	Fathatain		Ada

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :





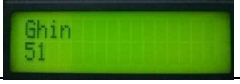
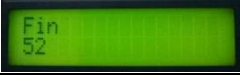




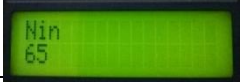





No.	Huruf Hijaiyah	Tanda Baca Huruf	Tampilan LCD	Suara
16.	ط	Fathatain		Ada
17.	ظ	Fathatain		Ada
18.	ع	Fathatain		Ada
19.	ع	Fathatain		Ada
20.	ف	Fathatain		Ada
21.	ق	Fathatain		Ada
22.	ك	Fathatain		Ada
23.	ل	Fathatain		Ada
24.	م	Fathatain		Ada
25.	ن	Fathatain		Ada
26.	و	Fathatain		Ada
27.	ه	Fathatain		Ada
28.	لأ	Fathatain		Ada
29.	ء	Fathatain		Ada
30.	ي	Fathatain		Ada

Tabel 28. Data Pengujian Kombinasi Huruf dan Tanda Baca Kasrotain




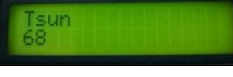

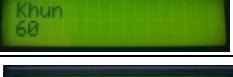








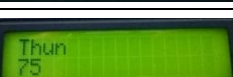

No.	Huruf Hijaiyah	Tanda Baca Huruf	Tampilan LCD	Suara
1.	ا	Kasrotain		Ada
2.	ب	Kasrotain		Ada
3.	ت	Kasrotain		Ada
4.	ث	Kasrotain		Ada
5.	ج	Kasrotain		Ada
6.	ح	Kasrotain		Ada
7.	خ	Kasrotain		Ada
8.	د	Kasrotain		Ada
9.	ذ	Kasrotain		Ada
10.	ر	Kasrotain		Ada
11.	ز	Kasrotain		Ada
12.	س	Kasrotain		Ada
13.	ش	Kasrotain		Ada
14.	ص	Kasrotain		Ada

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :















No.	Huruf Hijaiyah	Tanda Baca Huruf	Tampilan LCD	Suara
15.	ضِ	Kasrotain		Ada
16.	طِ	Kasrotain		Ada
17.	ظِ	Kasrotain		Ada
18.	عِ	Kasrotain		Ada
19.	غِ	Kasrotain		Ada
20.	فِ	Kasrotain		Ada
21.	قِ	Kasrotain		Ada
22.	كِ	Kasrotain		Ada
23.	لِ	Kasrotain		Ada
24.	مِ	Kasrotain		Ada
25.	نِ	Kasrotain		Ada
26.	وِ	Kasrotain		Ada
27.	هِ	Kasrotain		Ada
28.	لاِ	Kasrotain		Ada
29.	ءِ	Kasrotain		Ada
30.	يِ	Kasrotain		Ada

Tabel 29. Data Pengujian Kombinasi Huruf dan Tanda Baca Domatain

No.	Huruf Hijaiyah	Tanda Baca Huruf	Tampilan LCD	Suara
1.	ا	Domatain		Ada
2.	ب	Domatain		Ada
3.	ت	Domatain		Ada
4.	ث	Domatain		Ada
5.	ج	Domatain		Ada
6.	ح	Domatain		Ada
7.	خ	Domatain		Ada
8.	د	Domatain		Ada
9.	ذ	Domatain		Ada
10.	ر	Domatain		Ada
11.	ز	Domatain		Ada
12.	س	Domatain		Ada
13.	ش	Domatain		Ada
14.	ص	Domatain		Ada
15.	ض	Domatain		Ada
16.	ط	Domatain		Ada

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Huruf Hijaiyah	Tanda Baca Huruf	Tampilan LCD	Suara
17.	ظ	Domatain		Ada
18.	ع	Domatain		Ada
19.	ع	Domatain		Ada
20.	ف	Domatain		Ada
21.	ق	Domatain		Ada
22.	ك	Domatain		Ada
23.	ل	Domatain		Ada
24.	م	Domatain		Ada
25.	ن	Domatain		Ada
26.	و	Domatain		Ada
27.	ه	Domatain		Ada
28.	ي	Domatain		Ada
29.	ء	Domatain		Ada
30.	ي	Domatain		Ada

Dari tabel pengujian kombinasi huruf dan tanda baca huruf ini merupakan data hasil pengujian kombinasi antara ke-30 tombol huruf hijaiyah dengan ke-6 tanda baca huruf. Pengujian dilakukan dengan cara (*shift*) menekan lama tombol tanda baca, kemudian menekan salah satu huruf hijaiyah, maka suara yang dihasilkan akan berbeda dengan huruf

hijaiyah yang tidak dikombinasikan dengan tanda baca huruf. Setiap tombol yang ditekan lama akan menampilkan hasil *output* yang berbeda - beda. Tampilan *output* pada LCD yang merepresentasikan data hasil pengujian menunjukkan kesesuaian dengan tombol yang ditekan.

d. Pengukuran Kecepatan Tombol *Input* dan *Output* Suara

Pengukuran kecepatan pada *output* suara ini bertujuan untuk mengetahui berapa jeda waktu yang dibutuhkan untuk menampilkan suara disaat tombol *input* huruf ditekan. Pengukuran ini dilakukan dengan cara mengatur tombol yang akan ditekan dengan jarak waktu suara huruf berbunyi. Setelah melakukan uji coba pengukuran kecepatan *output* suara maka diperoleh hasil. Tabel di bawah ini merupakan data hasil dari pengukuran kecepatan pada *output* suara huruf hijaiyah.

Tabel 30. Data Pengukuran Kecepatan Tombol *Input* Huruf hijaiyah dan *Output* Suara

No.	Huruf Hijaiyah	Kecepatan <i>Output</i> Suara/(detik)
1.	ا	1 detik
2.	ب	1 detik
3.	ت	2 detik
4.	ث	2 detik
5.	ج	1 detik
6.	ح	1 detik
7.	خ	2 detik
8.	د	1 detik
9.	ذ	2 detik

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Huruf Hijaiyah	Kecepatan <i>Output</i> Suara/(detik)
10.	ر	2 detik
11.	ز	2 detik
12.	س	2 detik
13.	ش	2 detik
14.	ص	2 detik
15.	ض	1 detik
16.	ط	1 detik
17.	ظ	2 detik
18.	ع	1 detik
19.	غ	2 detik
20.	ف	2 detik
21.	ق	1 detik
22.	ك	1 detik
23.	ل	2 detik
24.	م	1 detik
25.	ن	1 detik
26.	و	2 detik
27.	ه	1 detik
28.	لا	2 detik
29.	ء	2 detik
30.	ي	1 detik

Tabel 31. Data Pengukuran Kecepatan Tombol *Input* Tanda Baca Fathah dan *Output* Suara

No.	Huruf Hijaiyah	Tanda Baca Huruf	Kecepatan <i>Output</i> Suara/(detik)
1.	أَ	Fathah	2 detik
2.	بَ	Fathah	1 detik
3.	تَ	Fathah	2 detik
4.	ثَ	Fathah	2 detik
5.	جَ	Fathah	1 detik
6.	حَ	Fathah	1 detik
7.	خَ	Fathah	2 detik
8.	دَ	Fathah	2 detik
9.	ذَ	Fathah	1 detik
10.	رَ	Fathah	2 detik
11.	زَ	Fathah	1 detik
12.	سَ	Fathah	1 detik
13.	شَ	Fathah	2 detik
14.	صَ	Fathah	2 detik
15.	ضَ	Fathah	1 detik
16.	طَ	Fathah	1 detik
17.	ظَ	Fathah	2 detik
18.	عَ	Fathah	2 detik
19.	غَ	Fathah	2 detik
20.	فَ	Fathah	1 detik
21.	قَ	Fathah	2 detik
22.	كَ	Fathah	1 detik
23.	لَ	Fathah	1 detik
24.	مَ	Fathah	1 detik
25.	نَ	Fathah	1 detik
26.	وَ	Fathah	1 detik

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Huruf Hijaiyah	Tanda Baca Huruf	Kecepatan <i>Output</i> Suara/(detik)
27.	هـ	Fathah	1 detik
28.	و	Fathah	1 detik
29.	ز	Fathah	1 detik
30.	ح	Fathah	1 detik

Tabel 32. Data Pengukuran Kecepatan Tombol *Input* Tanda Baca Kasroh dan *Output* Suara

No.	Huruf Hijaiyah	Tanda Baca Huruf	Kecepatan <i>Output</i> Suara/(detik)
1.	ا	Kasroh	1 detik
2.	ب	Kasroh	1 detik
3.	ت	Kasroh	2 detik
4.	ث	Kasroh	1 detik
5.	ج	Kasroh	1 detik
6.	ح	Kasroh	2 detik
7.	خ	Kasroh	1 detik
8.	د	Kasroh	1 detik
9.	ذ	Kasroh	2 detik
10.	ر	Kasroh	2 detik
11.	ز	Kasroh	1 detik
12.	س	Kasroh	1 detik
13.	ش	Kasroh	2 detik
14.	ص	Kasroh	1 detik
15.	ض	Kasroh	1 detik
16.	ط	Kasroh	1 detik
17.	ظ	Kasroh	1 detik
18.	ع	Kasroh	1 detik

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Huruf Hijaiyah	Tanda Baca Huruf	Kecepatan <i>Output</i> Suara/(detik)
19.	غ	Kasroh	2 detik
20.	ف	Kasroh	1 detik
21.	ق	Kasroh	1 detik
22.	ك	Kasroh	1 detik
23.	ل	Kasroh	1 detik
24.	م	Kasroh	2 detik
25.	ن	Kasroh	2 detik
26.	و	Kasroh	1 detik
27.	ه	Kasroh	2 detik
28.	لا	Kasroh	1 detik
29.	ء	Kasroh	1 detik
30.	ي	Kasroh	1 detik

Tabel 33. Data Pengukuran Kecepatan Tombol *Input* Tanda Baca Domah dan *Output* Suara

No.	Huruf Hijaiyah	Tanda Baca Huruf	Kecepatan <i>Output</i> Suara/(detik)
1.	أ	Domah	1 detik
2.	ب	Domah	1 detik
3.	ت	Domah	1 detik
4.	ث	Domah	2 detik
5.	ج	Domah	1 detik
6.	ح	Domah	2 detik
7.	خ	Domah	1 detik
8.	د	Domah	1 detik
9.	ذ	Domah	1 detik
10.	ر	Domah	1 detik

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Huruf Hijaiyah	Tanda Baca Huruf	Kecepatan <i>Output</i> Suara/(detik)
11.	زُ	Domah	2 detik
12.	سُ	Domah	2 detik
13.	شُ	Domah	1 detik
14.	صُ	Domah	1 detik
15.	ضُ	Domah	2 detik
16.	طُ	Domah	1 detik
17.	ظُ	Domah	1 detik
18.	عُ	Domah	1 detik
19.	غُ	Domah	2 detik
20.	فُ	Domah	1 detik
21.	قُ	Domah	2 detik
22.	كُ	Domah	1 detik
23.	لُ	Domah	1 detik
24.	مُ	Domah	1 detik
25.	نُ	Domah	2 detik
26.	وُ	Domah	1 detik
27.	هُ	Domah	1 detik
28.	لَا	Domah	2 detik
29.	ءُ	Domah	1 detik
30.	يُ	Domah	1 detik

Tabel 34. Data Pengukuran Kecepatan Tombol *Input* Tanda Baca Fathatain dan *Output* Suara

No.	Huruf Hijaiyah	Tanda Baca Huruf	Kecepatan <i>Output</i> Suara/(detik)
1.	أ	Fathatain	2 detik
2.	ب	Fathatain	2 detik
3.	ت	Fathatain	1 detik
4.	ث	Fathatain	2 detik
5.	ج	Fathatain	2 detik
6.	ح	Fathatain	1 detik
7.	خ	Fathatain	1 detik
8.	د	Fathatain	2 detik
9.	ذ	Fathatain	1 detik
10.	ر	Fathatain	1 detik
11.	ز	Fathatain	2 detik
12.	س	Fathatain	1 detik
13.	ش	Fathatain	2 detik
14.	ص	Fathatain	2 detik
15.	ض	Fathatain	1 detik
16.	ط	Fathatain	2 detik
17.	ظ	Fathatain	1 detik
18.	ع	Fathatain	2 detik
19.	غ	Fathatain	1 detik
20.	ف	Fathatain	2 detik
21.	ق	Fathatain	1 detik
22.	ك	Fathatain	2 detik
23.	ل	Fathatain	1 detik
24.	م	Fathatain	2 detik
25.	ن	Fathatain	1 detik
26.	و	Fathatain	2 detik

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Huruf Hijaiyah	Tanda Baca Huruf	Kecepatan <i>Output</i> Suara/(detik)
27.	هـ	Fathatain	2 detik
28.	لـ	Fathatain	1 detik
29.	ءـ	Fathatain	1 detik
30.	يـ	Fathatain	1 detik

Tabel 35. Data Pengukuran Kecepatan Tombol *Input* Tanda Baca Kasrotain dan *Output* Suara

No.	Huruf Hijaiyah	Tanda Baca Huruf	Kecepatan <i>Output</i> Suara/(detik)
1.	اـ	Kasrotain	1 detik
2.	بـ	Kasrotain	2 detik
3.	تـ	Kasrotain	2 detik
4.	ثـ	Kasrotain	1 detik
5.	جـ	Kasrotain	1 detik
6.	حـ	Kasrotain	2 detik
7.	خـ	Kasrotain	1 detik
8.	دـ	Kasrotain	1 detik
9.	ذـ	Kasrotain	2 detik
10.	رـ	Kasrotain	1 detik
11.	زـ	Kasrotain	1 detik
12.	سـ	Kasrotain	1 detik
13.	شـ	Kasrotain	1 detik
14.	صـ	Kasrotain	2 detik
15.	ضـ	Kasrotain	2 detik
16.	طـ	Kasrotain	2 detik
17.	ظـ	Kasrotain	1 detik
18.	عـ	Kasrotain	2 detik

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Huruf Hijaiyah	Tanda Baca Huruf	Kecepatan <i>Output</i> Suara/(detik)
19.	غ	Kasrotain	1 detik
20.	ف	Kasrotain	1 detik
21.	ق	Kasrotain	1 detik
22.	ك	Kasrotain	1 detik
23.	ل	Kasrotain	1 detik
24.	م	Kasrotain	1 detik
25.	ن	Kasrotain	1 detik
26.	و	Kasrotain	1 detik
27.	ه	Kasrotain	1 detik
28.	لا	Kasrotain	1 detik
29.	ء	Kasrotain	1 detik
30.	ي	Kasrotain	1 detik

Tabel 36. Data Pengukuran Kecepatan Tombol *Input* Tanda Baca Domatain dan *Output* Suara

No.	Huruf Hijaiyah	Tanda Baca Huruf	Kecepatan <i>Output</i> Suara/(detik)
1.	أ	Domatain	2 detik
2.	ب	Domatain	1 detik
3.	ت	Domatain	2 detik
4.	ث	Domatain	1 detik
5.	ج	Domatain	1 detik
6.	ح	Domatain	2 detik
7.	خ	Domatain	2 detik
8.	د	Domatain	2 detik
9.	ذ	Domatain	1 detik
10.	ر	Domatain	1 detik

Lanjut ke tabel berikutnya :

Sambungan tabel sebelumnya :

No.	Huruf Hijaiyah	Tanda Baca Huruf	Kecepatan <i>Output</i> Suara/(detik)
11.	ز	Domatain	2 detik
12.	س	Domatain	1 detik
13.	ش	Domatain	2 detik
14.	ص	Domatain	1 detik
15.	ض	Domatain	2 detik
16.	ط	Domatain	2 detik
17.	ظ	Domatain	2 detik
18.	ع	Domatain	2 detik
19.	غ	Domatain	1 detik
20.	ف	Domatain	1 detik
21.	ق	Domatain	2 detik
22.	ك	Domatain	1 detik
23.	ل	Domatain	2 detik
24.	م	Domatain	1 detik
25.	ن	Domatain	2 detik
26.	و	Domatain	1 detik
27.	ه	Domatain	1 detik
28.	لا	Domatain	1 detik
29.	ء	Domatain	1 detik
30.	ي	Domatain	2 detik

C. Pembahasan

1. Pengukuran Tegangan Kerja *Output* Huruf Hijaiyah

Pengukuran tegangan yang dipakai secara keseluruhan menggunakan tegangan 12VDC, namun *supply* tegangan yang dipakai hanya 5VDC. Dari hasil pengukuran tegangan menunjukkan bahwa media pembelajaran huruf hijaiyah

dapat bekerja dengan menggunakan tegangan minimal 5 *volt*. Jika tegangan yang digunakan di bawah 5 *volt*, maka media pembelajaran huruf hijaiyah masih dapat bekerja, namun tidak maksimal. Bunyi huruf terdengar tidak jelas dan tampilan tanda baca pada LCD redup. Bila diberi tegangan di atas 12 *volt* dapat mengakibatkan IC regulator cepat panas. Dengan demikian tegangan efektif yang digunakan sebagai *supply* tegangan sebesar 4,5 -12*volt*.

Pada pengukuran tegangan *supply* pada IC ATmega16 menunjukkan hasil pengukuran sebesar 5 *volt*. Tegangan 5 *volt* juga terbaca pada pengukuran tegangan *supply* untuk IC TDA 7052. Dengan demikian, IC regulator 7805 dapat bekerja dengan baik menghasilkan keluaran sebesar 5 *volt*.

Pengukuran tegangan terakhir adalah mengukur besarnya tegangan sumber dari *micro sd*. Tegangan sumber *mikro sd* (kaki no. 4) dapat diatur dengan memutar potensiometer yang terpasang sebagai pembagi tegangan. Hasil pengukuran menunjukkan bahwa *mikro sd* dapat terbaca oleh mikrokontroler (dapat mengeluarkan suara) pada tegangan minimum sebesar 3,2 *volt*. Pada tegangan di atas 2,5 *volt*, *mikro sd* akan terbaca akan tetapi dioda zener yang membatasi tegangan tersebut akan cepat panas. Dengan demikian, *mikro sd* akan bekerja efektif bila menggunakan sumber tegangan antara 1,8 hingga 3,4 *volt*.

2. Pengujian Tombol Huruf Hijaiyah

Dari hasil pengujian diketahui bahwa huruf hijaiyah terbagi atas 30 tombol huruf. Pengujian tombol huruf hijaiyah ini dimaksudkan untuk mengetahui apakah tombol huruf dapat berfungsi dengan baik atau tidak. Setiap tombol memiliki *output* suara dan tanda baca pada LCD yang berbeda-beda. Hal ini terlihat dari

data sampel ke-30 huruf hijaiyah pada Tabel 14 merupakan hasil pengujian dari tiap - tiap tombol huruf.

Tombol huruf hijaiyah dapat berjalan sesuai dengan fungsinya, namun jika tombol huruf menekan terlalu lama maka akan terjadi pengiriman data yang secara terus menerus sampai tombol huruf itu dilepaskan. Hal ini dikarenakan setiap tombol huruf yang ditekan akan mengirimkan data berupa kode angka yang akan dikirimkan kedalam *chip* mikrokontroler.

3. Pengujian Kombinasi Huruf dan Tanda Baca Huruf

Dari hasil pengujian tombol huruf hijaiyah dapat berjalan dengan baik, hal ini dapat dilihat dari tabel 14 pada pengujian sebelumnya. Setelah dikatakan baik, hasil dari pengujian tombol huruf hijaiyah maka langkah selanjutnya akan dilakukan pengujian dengan cara mengkomponasi tombol huruf hijaiyah dengan tombol tanda baca huruf.

Tombol huruf hijaiyah yang akan dikombinasikan terdiri dari 30 huruf, sedangkan tombol tanda baca huruf terdiri dari 6 tanda baca yaitu *fathah*, *kasroh*, *domah*, *fathatain*, *kasrotain* dan *domatain*. Jumlah huruf yang akan diujikan terdiri 180 sampel huruf. Pengujian dilakukan dengan cara menekan tombol tanda baca huruf (tanpa lepas), kemudian memilih satu per satu tombol huruf hijaiyah maka *output* pada *speaker* akan mengeluarkan bunyi huruf dan LCD akan menampilkan bacaan huruf hijaiyah, setelah itu bacaan huruf pada LCD akan kembali ke bentuk awal huruf. Hal ini dimaksudkan agar pengguna dapat mengingat kembali bacaan huruf hijaiyah sebelum menggunakan tanda baca huruf.

4. Pengukuran Kecepatan Tombol *Input* dan *Output* Suara

Setiap tombol huruf hijaiyah memiliki kecepatan pengiriman data yang berbeda - beda, hal ini disebabkan karena banyaknya *input* tombol yang digunakan serta banyaknya *file* suara yang disimpan didalam *micro sd* sehingga mengakibatkan proses pengiriman data dari tiap - tiap tombol membutuhkan waktu. Pengukuran dilakukan dengan cara menghitung kecepatan tombol huruf yang telah dieksekusi dan membandingkan jarak jeda antara tombol huruf satu dengan yang lainnya. Dari hasil eksekusi didapatkan *output* suara dengan *delay* yang berbeda - beda. Perbedaan jarak jeda itu dikarenakan faktor bacaan huruf yang panjang, sehingga *delay* yang dibutuhkan untuk mengatur jarak harus menyesuaikan bacaan hurufnya.

D. Unjuk Kerja Alat

Proses kerja pada media pengenalan huruf hijaiyah ini membutuhkan tegangan 12 VDC untuk mengoperasikan seluruh rangkaian sistem. Rangkaian itu terdiri dari rangkaiain sistem minimum ATmega16, rangkaian penguat OP-AMP, rangkaian display LCD. Unjuk kerja dari media pembelajaran huruf hijaiyah dilakukan dengan cara menekan tombol ON/OFF kemudian mengatur besar kecilnya suara yang dibutuhkan dengan cara mengatur *volume* suara. Pada tampilan awal LCD akan muncul karakter huruf “MEDIA BELAJAR HURUF HIJAIYAH” kemudian tekan tombol huruf hijaiyah maka keluaran display LCD akan menampilkan karakter bacaan huruf dan keluaran *Speaker* akan mengeluarkan suara huruf. Setiap tombol huruf memiliki kode tombol yang berbeda - beda, sehingga diharapkan pengguna tidak menekan tombol lebih dari

satu. Huruf hijaiyah terdiri dari 30 huruf yang akan dikombinasikan dengan tanda baca huruf yaitu *fathah*, *kasroh*, *domah*, *fathatain*, *kasrotain* dan *domatain*. Hal ini dilakukan dengan cara menekan salah satu tombol tanda baca huruf tanpa lepas (*shift*) kemudian memilih salah satu dari ke-30 huruf hijaiyah, maka *output* display pada LCD akan menampilkan karakter huruf dan *speaker* akan mengeluarkan suara huruf. Pada tampilan display LCD dan suara memiliki perbedaan yang signifikan setelah huruf di kombinasikan dengan ke-6 tanda baca huruf hijaiyah.

BAB V

KESIMPULAN DAN SARAN

Dari pembuatan proyek akhir ini dapat diperoleh kesimpulan sebagai berikut :

A. Kesimpulan

1. Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16 ini menggunakan perangkat keras (*Hardware*) yang terdiri dari Rangkaian Sistem Minimum ATmega16, Rangkaian Penguat OP-AMP, Rangkaian Catu Daya, Rangkaian Display LCD. Rangkaian Sistem Minimum ATmega16 ini difungsikan sebagai rangkaian pengendali utama dari keseluruhan rangkaian. Sedangkan rangkaian catu daya digunakan untuk men-supply tegangan keseluruh rangkaian. Rangkaian penguat OP-AMP difungsikan sebagai penguat suara, agar suara yang dihasilkan tidak pecah dan tidak menimbulkan banyak *noise*. Rangkaian display LCD difungsikan untuk menampilkan hasil keluaran dari *input* tombol huruf hijaiyah yang ditekan.
2. Perangkat lunak (*software*) yang digunakan terdiri dari *software* CVAVR_Evaluation dan Proteus Professional 7. Pada *Software* CVAVR_Evaluation difungsikan untuk membuat program pengendali (input, pemroses dan output), mengedit program dan men-download program pada Sistem Minimum ATmega16. Untuk *software* Proteus professional difungsikan untuk membuat desain rangkaian dan desain *layout* PCB.
3. Unjuk kerja dari pembuatan Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis

Mikrokontroler ATmega16 yaitu dapat bekerja dengan baik dan berjalan sesuai dengan fungsinya. Prinsip kerja dari alat ini menggunakan keyboard serial PS2 sebagai media belajar huruf hijaiyah. Keyboard dapat dikombinasikan dengan sistem minimum ATmega16, hal ini dilakukan dengan cara mengetahui kode-kode tombol pada keyboard yang akan dimanfaatkan untuk mengirim data kode melalui IC mikrokontroler ATmega16. Setelah terkirim kemudian memanggil file data yang disimpan di dalam *micro sd* melalui modul *MP3 Player* dan hasilnya display LCD akan menampilkan karakter berupa bacaan dan suara dari speaker.

B. Keterbatasan Alat

Pada perancangan Media Pengenalan Huruf Hijaiyah untuk Taman Kanak - Kanak dengan *Output* Suara dan Tanda Baca Berbasis Mikrokontroler ATmega16 ini tidak lepas dari keterbatasan alat sehingga dengan terciptanya alat ini dapat dijadikan acuan untuk kemajuan dan perkembangan teknologi dikemudian hari. Berikut ini adalah beberapa keterbatasan dari alat yaitu :

1. Alat pembelajaran huruf hijaiyah ini belum tersedianya fitur untuk menampilkan karakter huruf arab pada LCD.
2. Rangkaian penguat OP-AMP dan *Speaker* yang kurang sempurna mengakibatkan *output* suara terkadang terdengar kurang jelas.

C. Saran

Pada pembuatan proyek akhir ini tidak terlepas dari kekurangan, sehingga diperlukan saran yang membangun untuk perkembangan proyek akhir ini, sebagai berikut :

1. Perlu adanya pengembangan media pembelajaran huruf hijaiyah dimana *output* dapat menampilkan karakter huruf arab, agar pengguna dapat membedakan bentuk karakter huruf pada tombol *input* dengan *output* penampil huruf pada LCD.
2. Pada pengembangan media pembelajaran huruf hijaiyah selanjutnya diharapkan dapat menambahkan fitur - fitur yang mendukung seperti pembuatan *game* guna sebagai latihan untuk mengenal huruf hijaiyah.

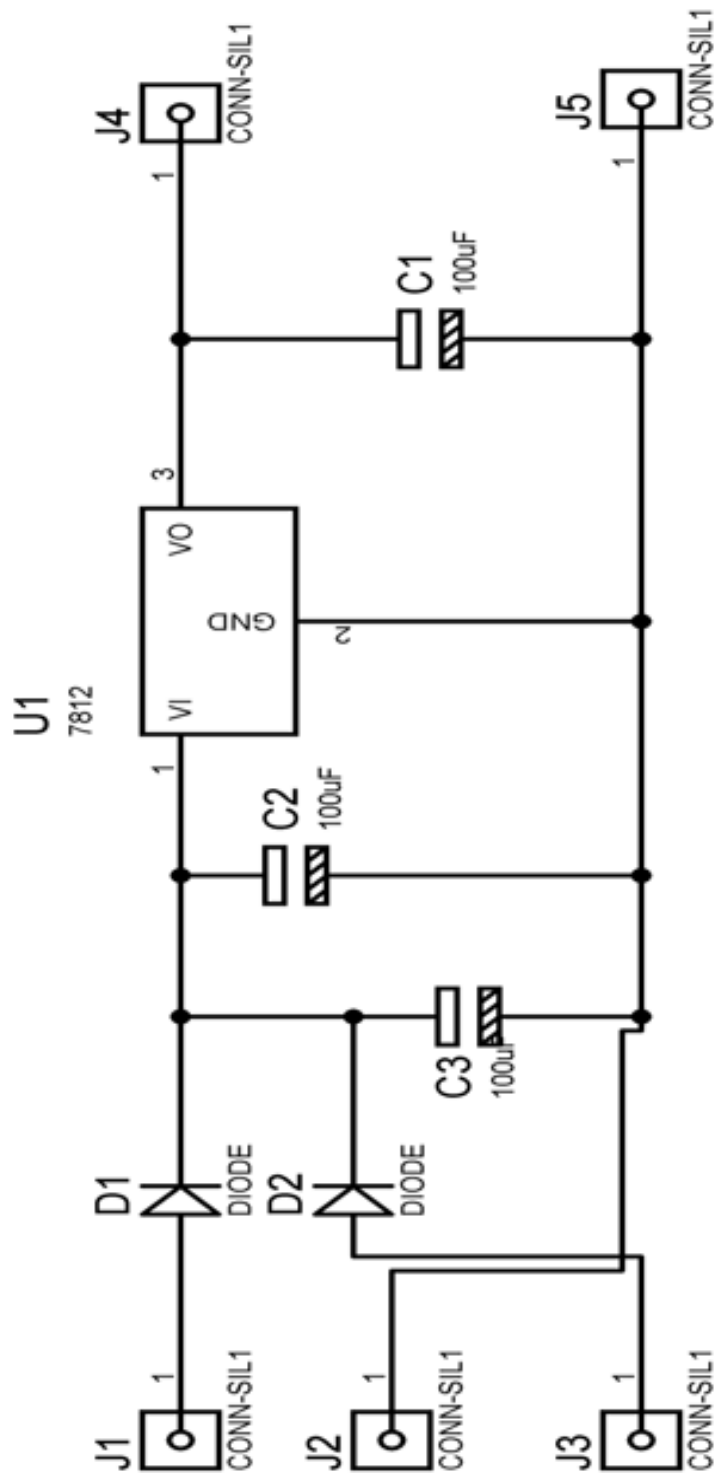
DAFTAR PUSTAKA

- Andrianto, Heri. 2008. *Pemrograman Mikrokontroler AVR Atmega16*. Bandung: Penerbit Informatika bandung.
- Arsyad, Azhar. 2002. *Media pembelajaran*. Jakarta: Penerbit PT. Raja Grafindo Persada.
- Atmel Corporation. 2003. *Datasheet ATmega16*. Diambil dari :<http://www.atmel.com/Images/doc2466.pdf>. pada tanggal 10 januari 2014.
- Chan. (2014). *FatFs – Generic Fat File System Module*. Diakses dari <http://elm-chan.org>. pada tanggal 16 Desember 2013.
- Elyas, Ali Mustahib, dkk. 2011. *Pendidikan Agama Islam*. Jakarta: Pusat Kurikulum dan Pembukuan, Kementrian Pendidikan Nasional.
- Engineersgrage (2014). Diakses dari: <http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet>. pada tanggal 23 Juli 2014.
- Gerlach, V.G. dan Ely, D.P. 1971. *Teaching and Media. A Systematic Approach*. Englewood Cliffs: Prentice-Hall, Inc.
- Hadi, Solihul. 2008. *Mengenal Mikrokontroler ATmega16*. Diakses dari: <http://ilmukomputer.org/wp-content/uploads/2008/08/sholihul-atmega16.pdf>. Pada tanggal 25 Oktober 2013.
- Muttaqien, Zaenal. (2009). *Mesin Ketik Huruf Braille Elektronik Berbasis Mikrokontroler Atmega16*. Tugas Akhir Universitas Negeri Yogyakarta 2012.
- PHILIPS. (1994). *DATASHEET TDA7052*. Diakses dari: <http://www.alldatasheet.com>. Pada tanggal 3 Januari 2014.
- Puspito Hendro. 1983. *Sosiologi Agama*. Yogyakarta : Penerbit Kanisius.
- Safrudin, Cholis. 2011. *Proteus Introduction*. Diakses dari: <http://ydlchs.files.wordpress.com/2014/03/28-introduction-proteus.pdf>. Pada tanggal 28 Maret 2014.
- Sidiq, Sapiuddin, dkk. 2012. *Modul Al-Qur'an*. Jakarta: PT. Makhtubullah. Halaman 4.
- Syaltout, Mahmud. 1983. *Islam sebagai Aqidah dan Syar'iah*. Terj. Bustami A. Gani & B. Hamdany Ali. Jakarta: Bulan Bintang. Jilid 1, 2, & 3.

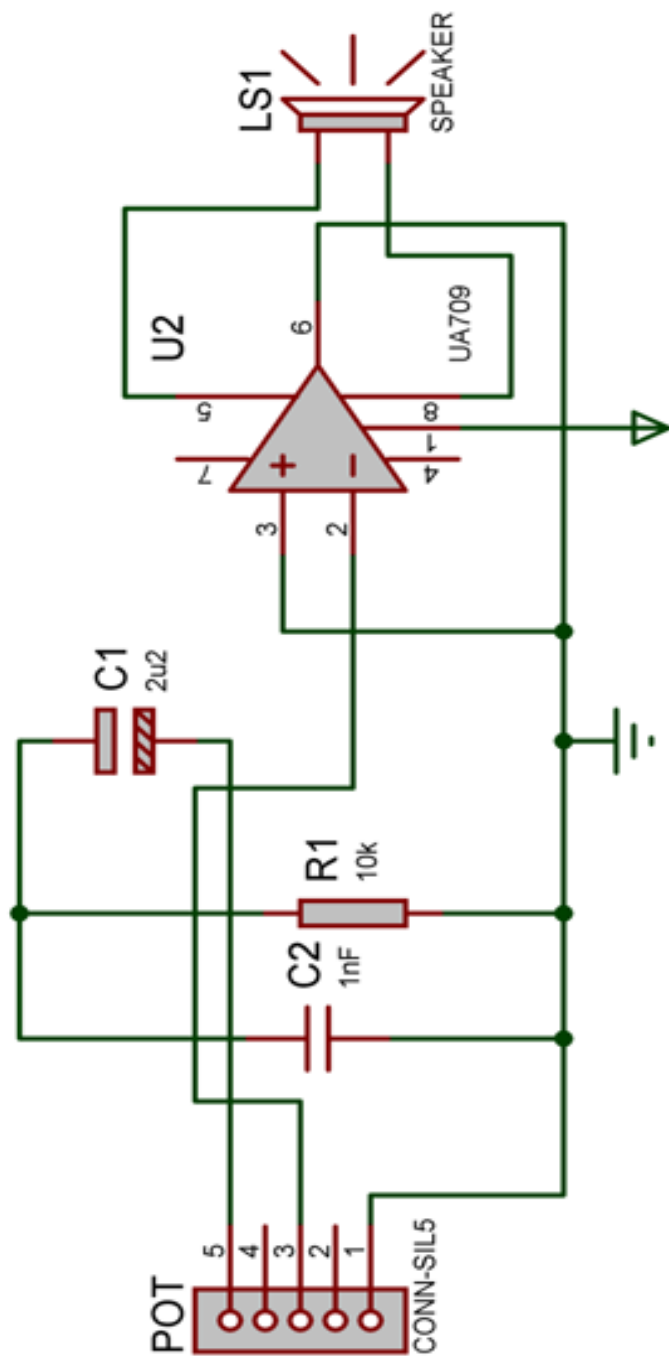
Syaodih, Ernawulan. (2014). *Perkembangan Anak di Taman Kanak-Kanak*. Tugas Akhir Skripsi. Universitas Pendidikan Indonesia 2014.

Wytronic. (2014). *Modul MP3 Player tipe WT5001M02-28P*. Diakses dari: <http://waytronic.ru/waytronic15.htm>. Pada tanggal 24 Juli 2014.

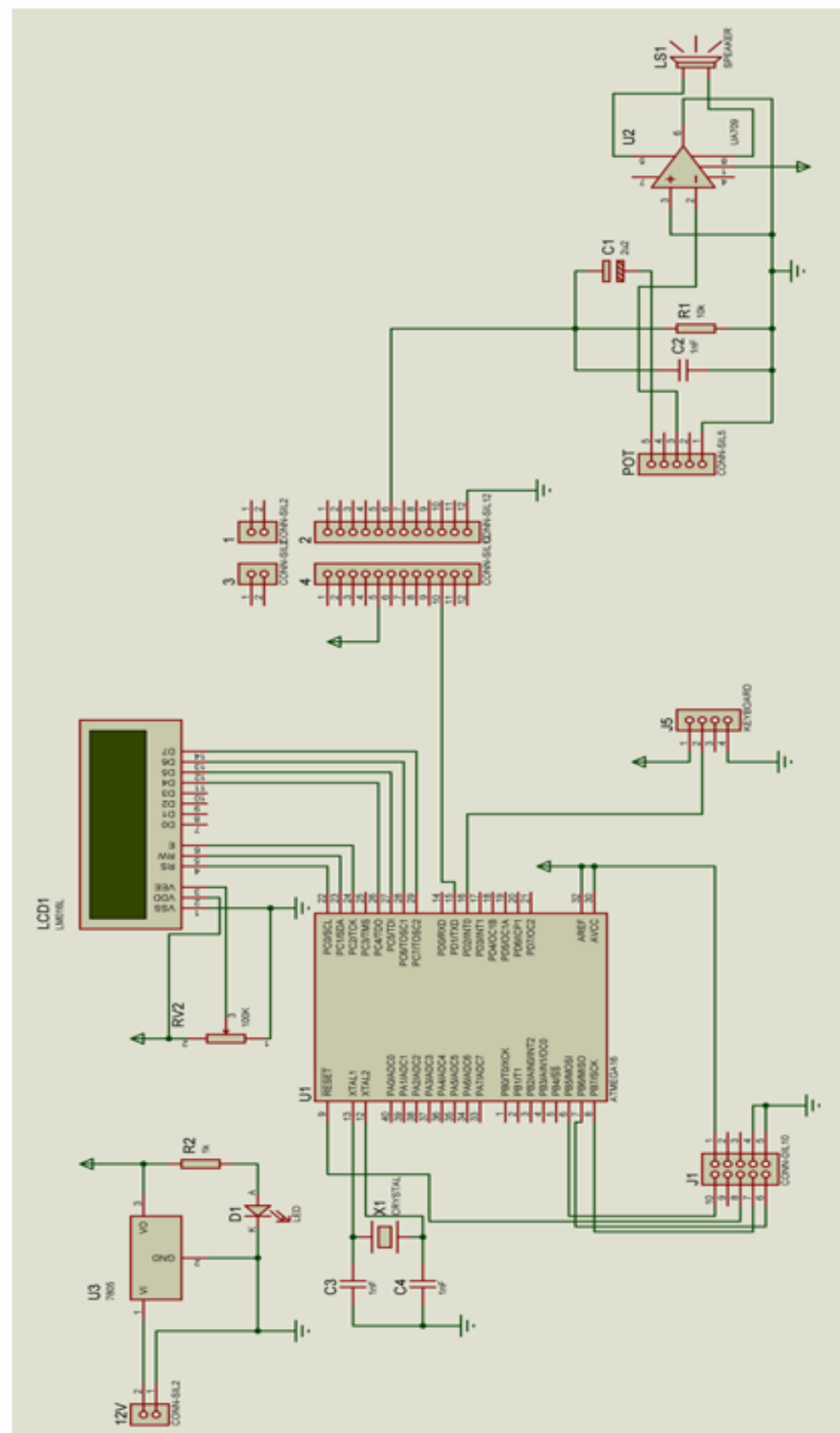




Skema Rangkaian Catu Daya			keterangan	
FT UNY	SKALA :	DIG Mulyana	A4	NO. 1
	DIP. BU RATNA	DIS. BU RATNA	Nim :10507131013	



Skema Rangkaian Amplifier			keterangan	
FT UNY	SKALA :		A4	NO.2
	DIP.BU RATNA	DIG MUYANA DIS.BU RATNA	Nim :10507131013	



PROGRAM

/*****

This program was produced by the

CodeWizardAVR V2.05.3 Standard

Automatic Program Generator

© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 1/6/2014

Author : PerTic@n

Company : If You Like This Software,Buy It

Comments:

Chip type : ATmega16A

Program type : Application

AVR Core Clock frequency: 12.000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

*****/

```
#include <mega16.h>
```

```
#include <stdio.h>
```

```
#include <delay.h>
```

```
#include <alcd.h>
```

```

// Enter the data port here

static unsigned int count=11;
static unsigned char data =0;
static unsigned char is_up=0;
unsigned char buf[33],a,temp;
int x=0;

// External Interrupt 0 service routine

interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here
if (count > 2 && count <11)           // Read the data bits,neglecting the start bit
and the stop and parity bits
{
data = (data >> 1);
if(PIND & 8)
{data = data | 0x80;x=1;} // Store a '1'
}

if(--count ==0 ) // All bits received
{
count = 11;
PORTA=data;
data=0;

}
}

void play(int number)
{

putchar(0x7E);
putchar(0x04);
putchar(0xA0);
putchar(number>>8);
putchar(number&0xff);
putchar(0x7E);
delay_ms(100);
}

void stop()
{

```

```

    putchar(0x7e);
    putchar(0x02);
    putchar(0xA4);
    putchar(0x7e);

}

void suara(int no)
{ if(x==1 && a>0){play(no);x=2;delay_ms(20);}
}

void hijaiyah()
{ is_up =0;
  { switch (a)
    { case 21 : lcd_gotoxy(0,0); lcd_putsf("Sin");suara(12);delay_ms(1000); break;
//deret pertama
    case 29 : lcd_gotoxy(0,0); lcd_putsf("Zai");suara(11);delay_ms(1000); break;
    case 36 : lcd_gotoxy(0,0); lcd_putsf("Ro");suara(10); delay_ms(1000); break;
    case 45 : lcd_gotoxy(0,0); lcd_putsf("Dzal");suara(9);delay_ms(1000); break;
    case 44 : lcd_gotoxy(0,0); lcd_putsf("Dal");suara(8);delay_ms(1000); break;
    case 53 : lcd_gotoxy(0,0); lcd_putsf("Kho");suara(7);delay_ms(1000); break;
    case 60 : lcd_gotoxy(0,0); lcd_putsf("Kha");suara(6);delay_ms(1000); break;
    case 67 : lcd_gotoxy(0,0); lcd_putsf("Jim");suara(5);delay_ms(1000); break;
    case 68 : lcd_gotoxy(0,0); lcd_putsf("Tsa");suara(4);delay_ms(1000); break;
    case 77 : lcd_gotoxy(0,0); lcd_putsf("Ta");suara(3);delay_ms(1000); break;
    case 84 : lcd_gotoxy(0,0); lcd_putsf("Ba");suara(2); delay_ms(1000); break;
    case 91 : lcd_gotoxy(0,0); lcd_putsf("Alif");suara(1);delay_ms(1000); break;

    case 28 : lcd_gotoxy(0,0); lcd_putsf("Mim");suara(24);delay_ms(1100);
break; //deret kedua
    case 27 : lcd_gotoxy(0,0); lcd_putsf("Lam");suara(23);delay_ms(1100);
break;
    case 35 : lcd_gotoxy(0,0); lcd_putsf("Kaf");suara(22);delay_ms(1100); break;
    case 43 : lcd_gotoxy(0,0); lcd_putsf("Qaf");suara(21);delay_ms(1100); break;
    case 52 : lcd_gotoxy(0,0); lcd_putsf("Fa"); suara(20);delay_ms(1100); break;
    case 51 : lcd_gotoxy(0,0); lcd_putsf("Ghain");suara(19);delay_ms(1200);
break;
    case 59 : lcd_gotoxy(0,0); lcd_putsf("Ain");suara(18);delay_ms(1200); break;
    case 66 : lcd_gotoxy(0,0); lcd_putsf("Zdho");suara(17);delay_ms(1200);
break;
    case 75 : lcd_gotoxy(0,0); lcd_putsf("Tho");suara(16);delay_ms(1100); break;
    case 76 : lcd_gotoxy(0,0); lcd_putsf("Dhod");suara(15);delay_ms(1200);
break;
    case 82 : lcd_gotoxy(0,0); lcd_putsf("Shod");suara(14);delay_ms(1200);
break;
    case 93 : lcd_gotoxy(0,0); lcd_putsf("Syin");suara(13);delay_ms(1200);
break;

```



```

        case 33 : lcd_gotoxy(0,0); lcd_putsf("Ya");suara(30);delay_ms(1100);break;
//deret kedua
        case 42 : lcd_gotoxy(0,0);
lcd_putsf("HAMZAH");suara(29);delay_ms(1200);break;
        case 50 : lcd_gotoxy(0,0); lcd_putsf("LAM
ALIF");suara(28);delay_ms(1300);break;
        case 49 : lcd_gotoxy(0,0); lcd_putsf("Ha");suara(27);delay_ms(1100); break;
        case 58 : lcd_gotoxy(0,0); lcd_putsf("Wau");suara(26);delay_ms(1200);
break;
        case 65 : lcd_gotoxy(0,0); lcd_putsf("Nun");suara(25);delay_ms(1100); break;

        case 41 : lcd_gotoxy(0,0); lcd_putsf("MEDIA BELAJAR");lcd_gotoxy(0,1);
lcd_putsf("HURUF HIJAIYAH"); break;

    }
}

}

void fathah()
{
    { switch (a)
    {
        case 21 : lcd_gotoxy(0,0); lcd_putsf("Sa");suara(42);delay_ms(1000); break;
//deret pertama
        case 29 : lcd_gotoxy(0,0); lcd_putsf("Za");suara(41);delay_ms(1000); break;
        case 36 : lcd_gotoxy(0,0); lcd_putsf("Ro");suara(40);delay_ms(1000); break;
        case 45 : lcd_gotoxy(0,0); lcd_putsf("Dza");suara(39); delay_ms(1000); break;
        case 44 : lcd_gotoxy(0,0); lcd_putsf("Da");suara(38); delay_ms(1000); break;
        case 53 : lcd_gotoxy(0,0); lcd_putsf("Kho");suara(37); delay_ms(1000); break;
        case 60 : lcd_gotoxy(0,0); lcd_putsf("Kha");suara(36); delay_ms(1000); break;
        case 67 : lcd_gotoxy(0,0); lcd_putsf("Ja");suara(35); delay_ms(1000); break;
        case 68 : lcd_gotoxy(0,0); lcd_putsf("Tsa");suara(34);delay_ms(1000); break;
        case 77 : lcd_gotoxy(0,0); lcd_putsf("Ta");suara(33); delay_ms(1000); break;
        case 84 : lcd_gotoxy(0,0); lcd_putsf("Ba");suara(32); delay_ms(1000); break;
        case 91 : lcd_gotoxy(0,0); lcd_putsf("A");suara(31); delay_ms(1000); break;

        case 28 : lcd_gotoxy(0,0); lcd_putsf("Ma");suara(54); delay_ms(1000); break;
//deret kedua
        case 27 : lcd_gotoxy(0,0); lcd_putsf("La");suara(53); delay_ms(1000); break;
        case 35 : lcd_gotoxy(0,0); lcd_putsf("Ka");suara(52); delay_ms(1000); break;
        case 43 : lcd_gotoxy(0,0); lcd_putsf("Qa");suara(51);delay_ms(1000); break;
        case 52 : lcd_gotoxy(0,0); lcd_putsf("Fa");suara(50); delay_ms(1000);break;
        case 51 : lcd_gotoxy(0,0); lcd_putsf("Gho");suara(49); delay_ms(1000);break;
        case 59 : lcd_gotoxy(0,0); lcd_putsf("A");suara(48); delay_ms(1000); break;

```

```

        case 66 : lcd_gotoxy(0,0); lcd_putsf("Zdho");suara(47);delay_ms(1000);
break;
        case 75 : lcd_gotoxy(0,0); lcd_putsf("Tho");suara(46); delay_ms(1000);break;
        case 76 : lcd_gotoxy(0,0); lcd_putsf("Dho");suara(45); delay_ms(1000); break;
        case 82 : lcd_gotoxy(0,0); lcd_putsf("Sho");suara(44); delay_ms(1000);break;
        case 93 : lcd_gotoxy(0,0); lcd_putsf("Sya");suara(43);delay_ms(1000); break;

        case 33 : lcd_gotoxy(0,0); lcd_putsf("Ya");suara(60); delay_ms(1000);break;
//deret kedua
        case 42 : lcd_gotoxy(0,0); lcd_putsf("a");suara(59); delay_ms(1000); break;
        case 50 : lcd_gotoxy(0,0); lcd_putsf("laa");suara(58); delay_ms(1000); break;
        case 49 : lcd_gotoxy(0,0); lcd_putsf("Ha");suara(57); delay_ms(1000); break;
        case 58 : lcd_gotoxy(0,0); lcd_putsf("Wa");suara(56); delay_ms(1000); break;
        case 65 : lcd_gotoxy(0,0); lcd_putsf("Na");suara(55); delay_ms(1000); break;
    }
}

}
void kasroh()
{
    { switch (a)
    {
        case 21 : lcd_gotoxy(0,0); lcd_putsf("Si");suara(72); delay_ms(1000); break;
//deret pertama
        case 29 : lcd_gotoxy(0,0); lcd_putsf("Zi");suara(71);delay_ms(1000); break;
        case 36 : lcd_gotoxy(0,0); lcd_putsf("Ri");suara(70); delay_ms(1000); break;
        case 45 : lcd_gotoxy(0,0); lcd_putsf("Dzi");suara(69); delay_ms(1000); break;
        case 44 : lcd_gotoxy(0,0); lcd_putsf("Di");suara(68); delay_ms(1000); break;
        case 53 : lcd_gotoxy(0,0); lcd_putsf("Khi");suara(67); delay_ms(1000); break;
        case 60 : lcd_gotoxy(0,0); lcd_putsf("Khi");suara(66); delay_ms(1000); break;
        case 67 : lcd_gotoxy(0,0); lcd_putsf("Ji");suara(65); delay_ms(1000);break;
        case 68 : lcd_gotoxy(0,0); lcd_putsf("Tsi");suara(64);delay_ms(1000); break;
        case 77 : lcd_gotoxy(0,0); lcd_putsf("Ti");suara(63); delay_ms(1000); break;
        case 84 : lcd_gotoxy(0,0); lcd_putsf("Bi");suara(62); delay_ms(1000); break;
        case 91 : lcd_gotoxy(0,0); lcd_putsf("I");suara(61); delay_ms(1000); break;

        case 28 : lcd_gotoxy(0,0); lcd_putsf("Mi");suara(84); delay_ms(1000); break;
//deret kedua
        case 27 : lcd_gotoxy(0,0); lcd_putsf("Li");suara(83); delay_ms(1000); break;
        case 35 : lcd_gotoxy(0,0); lcd_putsf("Ki");suara(82); delay_ms(1000); break;
        case 43 : lcd_gotoxy(0,0); lcd_putsf("Qi");suara(81); delay_ms(1000); break;
        case 52 : lcd_gotoxy(0,0); lcd_putsf("Fi");suara(80);delay_ms(1000); break;
        case 51 : lcd_gotoxy(0,0); lcd_putsf("Ghi");suara(79);delay_ms(1000); break;
        case 59 : lcd_gotoxy(0,0); lcd_putsf("Ai");suara(78); delay_ms(1000); break;
        case 66 : lcd_gotoxy(0,0); lcd_putsf("Zdhi");suara(77); delay_ms(1000);
break;
        case 75 : lcd_gotoxy(0,0); lcd_putsf("Thi");suara(76); delay_ms(1000); break;

```

```

    case 76 : lcd_gotoxy(0,0); lcd_putsf("Dhi");suara(75); delay_ms(1000); break;
    case 82 : lcd_gotoxy(0,0); lcd_putsf("Shi");suara(74);delay_ms(1000); break;
    case 93 : lcd_gotoxy(0,0); lcd_putsf("Syi");suara(73); delay_ms(1000); break;

    case 33 : lcd_gotoxy(0,0); lcd_putsf("Yi");suara(90); delay_ms(1000); break;
//deret kedua
    case 42 : lcd_gotoxy(0,0); lcd_putsf("i");suara(89); delay_ms(1000); break;
    case 50 : lcd_gotoxy(0,0); lcd_putsf("lii");suara(88); delay_ms(1000); break;
    case 49 : lcd_gotoxy(0,0); lcd_putsf("Hi");suara(87); delay_ms(1000); break;
    case 58 : lcd_gotoxy(0,0); lcd_putsf("Wi");suara(86); delay_ms(1000); break;
    case 65 : lcd_gotoxy(0,0); lcd_putsf("Ni");suara(85); delay_ms(1000); break;

    }
}

}
void dhomah()
{
    { switch (a)
    {

        case 21 : lcd_gotoxy(0,0); lcd_putsf("Su");suara(102);delay_ms(1000); break;
//deret pertama
        case 29 : lcd_gotoxy(0,0); lcd_putsf("Zu");suara(101); delay_ms(1000); break;
        case 36 : lcd_gotoxy(0,0); lcd_putsf("Ru");suara(100);delay_ms(1000); break;
        case 45 : lcd_gotoxy(0,0); lcd_putsf("Dzu");suara(99);delay_ms(1000); break;
        case 44 : lcd_gotoxy(0,0); lcd_putsf("Du");suara(98); delay_ms(1000); break;
        case 53 : lcd_gotoxy(0,0); lcd_putsf("Khu");suara(97);delay_ms(1000); break;
        case 60 : lcd_gotoxy(0,0); lcd_putsf("Khu");suara(96); delay_ms(1000); break;
        case 67 : lcd_gotoxy(0,0); lcd_putsf("Ju");suara(95); delay_ms(1000); break;
        case 68 : lcd_gotoxy(0,0); lcd_putsf("Tsu");suara(94);delay_ms(1000); break;
        case 77 : lcd_gotoxy(0,0); lcd_putsf("Tu");suara(93); delay_ms(1000); break;
        case 84 : lcd_gotoxy(0,0); lcd_putsf("Bu");suara(92);delay_ms(1000); break;
        case 91 : lcd_gotoxy(0,0); lcd_putsf("U");suara(91); delay_ms(1000); break;

        case 28 : lcd_gotoxy(0,0); lcd_putsf("Mu");suara(114); delay_ms(1000);
break; //deret kedua
        case 27 : lcd_gotoxy(0,0); lcd_putsf("Lu");suara(113); delay_ms(1000); break;
        case 35 : lcd_gotoxy(0,0); lcd_putsf("Ku");suara(112); delay_ms(1000); break;
        case 43 : lcd_gotoxy(0,0); lcd_putsf("Qu");suara(111); delay_ms(1000);
break;
        case 52 : lcd_gotoxy(0,0); lcd_putsf("Fu");suara(110); delay_ms(1000);break;
        case 51 : lcd_gotoxy(0,0); lcd_putsf("Ghu");suara(109); delay_ms(1000);
break;
        case 59 : lcd_gotoxy(0,0); lcd_putsf("Au");suara(108); delay_ms(1000); break;
        case 66 : lcd_gotoxy(0,0); lcd_putsf("Zdhu");suara(107);delay_ms(1000);
break;

```

```

        case 75 : lcd_gotoxy(0,0); lcd_putsf("Thu");suara(106); delay_ms(1000);
break;
        case 76 : lcd_gotoxy(0,0); lcd_putsf("Dhu");suara(105);delay_ms(1000);
break;
        case 82 : lcd_gotoxy(0,0); lcd_putsf("Shu");suara(104); delay_ms(1000);
break;
        case 93 : lcd_gotoxy(0,0); lcd_putsf("Syu");suara(103); delay_ms(1000);
break;

        case 33 : lcd_gotoxy(0,0); lcd_putsf("Yu");suara(120);delay_ms(1000); break;
//deret kedua
        case 42 : lcd_gotoxy(0,0); lcd_putsf("u");suara(119); delay_ms(1000); break;
        case 50 : lcd_gotoxy(0,0); lcd_putsf("lu");suara(118);delay_ms(1000); break;
        case 49 : lcd_gotoxy(0,0); lcd_putsf("Hu");suara(117);delay_ms(1000); break;
        case 58 : lcd_gotoxy(0,0); lcd_putsf("Wu");suara(116); delay_ms(1000);
break;
        case 65 : lcd_gotoxy(0,0); lcd_putsf("Nu");suara(115); delay_ms(1000);
break;
    }
}

}
void fathah_tain()
{
    {switch (a)
    {
        case 21 : lcd_gotoxy(0,0); lcd_putsf("San");suara(132);delay_ms(1000);
break; //deret pertama
        case 29 : lcd_gotoxy(0,0); lcd_putsf("Zan");suara(131); delay_ms(1000);
break;
        case 36 : lcd_gotoxy(0,0); lcd_putsf("Ron");suara(130); delay_ms(1000);
break;
        case 45 : lcd_gotoxy(0,0); lcd_putsf("Dzan");suara(129);delay_ms(1000);
break;
        case 44 : lcd_gotoxy(0,0); lcd_putsf("Dan");suara(128); delay_ms(1000);
break;
        case 53 : lcd_gotoxy(0,0); lcd_putsf("Khon");suara(127); delay_ms(1000);
break;
        case 60 : lcd_gotoxy(0,0); lcd_putsf("Khan");suara(126); delay_ms(1000);
break;
        case 67 : lcd_gotoxy(0,0); lcd_putsf("Jan");suara(125); delay_ms(1000);
break;
        case 68 : lcd_gotoxy(0,0); lcd_putsf("Tsan");suara(124);delay_ms(1000);
break;
        case 77 : lcd_gotoxy(0,0); lcd_putsf("Tan");suara(123); delay_ms(1000);
break;

```

```

        case 84 : lcd_gotoxy(0,0); lcd_putsf("Ban");suara(122); delay_ms(1000);
break;
        case 91 : lcd_gotoxy(0,0); lcd_putsf("An");suara(121); delay_ms(1000); break;

        case 28 : lcd_gotoxy(0,0); lcd_putsf("Man");suara(144);delay_ms(1000);
break; //deret kedua
        case 27 : lcd_gotoxy(0,0); lcd_putsf("Lan");suara(143); delay_ms(1000);
break;
        case 35 : lcd_gotoxy(0,0); lcd_putsf("Kan");suara(142);
delay_ms(1000);break;
        case 43 : lcd_gotoxy(0,0); lcd_putsf("Qan");suara(141); delay_ms(1000);
break;
        case 52 : lcd_gotoxy(0,0); lcd_putsf("Fan");suara(140); delay_ms(1000);
break;
        case 51 : lcd_gotoxy(0,0); lcd_putsf("Ghon");suara(139);delay_ms(1000);
break;
        case 59 : lcd_gotoxy(0,0); lcd_putsf("An");suara(138); delay_ms(1000); break;
        case 66 : lcd_gotoxy(0,0); lcd_putsf("Zdhon");suara(137);delay_ms(1000);
break;
        case 75 : lcd_gotoxy(0,0); lcd_putsf("Thon");suara(136); delay_ms(1000);
break;
        case 76 : lcd_gotoxy(0,0); lcd_putsf("Dhon");suara(135);delay_ms(1000);
break;
        case 82 : lcd_gotoxy(0,0); lcd_putsf("Shon");suara(134); delay_ms(1000);
break;
        case 93 : lcd_gotoxy(0,0); lcd_putsf("Syan");suara(133);delay_ms(1000);
break;

        case 33 : lcd_gotoxy(0,0); lcd_putsf("Yan");suara(150);
delay_ms(1000);break; //deret kedua
        case 42 : lcd_gotoxy(0,0); lcd_putsf("an");suara(149); delay_ms(1000); break;
        case 50 : lcd_gotoxy(0,0); lcd_putsf("lan");suara(148); delay_ms(1000); break;
        case 49 : lcd_gotoxy(0,0); lcd_putsf("Han");suara(147); delay_ms(1000);
break;
        case 58 : lcd_gotoxy(0,0); lcd_putsf("Wan");suara(146); delay_ms(1000);
break;
        case 65 : lcd_gotoxy(0,0); lcd_putsf("Nan");suara(145); delay_ms(1000);
break;

    }
}
}
void kasroh_tain()
{
    {switch (a)
    {

```

```

        case 21 : lcd_gotoxy(0,0); lcd_putsf("Sin");suara(162); delay_ms(1000);
break;    //deret pertama
        case 29 : lcd_gotoxy(0,0); lcd_putsf("Zin");suara(161); delay_ms(1000);
break;
        case 36 : lcd_gotoxy(0,0); lcd_putsf("Rin");suara(160); delay_ms(1000);
break;
        case 45 : lcd_gotoxy(0,0); lcd_putsf("Dzin");suara(159);delay_ms(1000);
break;
        case 44 : lcd_gotoxy(0,0); lcd_putsf("Din");suara(158); delay_ms(1000);
break;
        case 53 : lcd_gotoxy(0,0); lcd_putsf("Khin");suara(157); delay_ms(1000);
break;
        case 60 : lcd_gotoxy(0,0); lcd_putsf("Khin");suara(156); delay_ms(1000);
break;
        case 67 : lcd_gotoxy(0,0); lcd_putsf("Jin");suara(155); delay_ms(1000); break;
        case 68 : lcd_gotoxy(0,0); lcd_putsf("Tsin");suara(154); delay_ms(1000);
break;
        case 77 : lcd_gotoxy(0,0); lcd_putsf("Tin");suara(153);delay_ms(1000);
break;
        case 84 : lcd_gotoxy(0,0); lcd_putsf("Bin");suara(152);delay_ms(1000);
break;
        case 91 : lcd_gotoxy(0,0); lcd_putsf("In");suara(151); delay_ms(1000);break;

        case 28 : lcd_gotoxy(0,0); lcd_putsf("Min");suara(174); delay_ms(1000);
break;    //deret kedua
        case 27 : lcd_gotoxy(0,0); lcd_putsf("Lin");suara(173);
delay_ms(1000);break;
        case 35 : lcd_gotoxy(0,0); lcd_putsf("Kin");suara(172); delay_ms(1000);
break;
        case 43 : lcd_gotoxy(0,0); lcd_putsf("Qin");suara(171); delay_ms(1000);
break;
        case 52 : lcd_gotoxy(0,0); lcd_putsf("Fin");suara(170);delay_ms(1000); break;
        case 51 : lcd_gotoxy(0,0); lcd_putsf("Ghin");suara(169); delay_ms(1000);
break;
        case 59 : lcd_gotoxy(0,0); lcd_putsf("Ain");suara(168);
delay_ms(1000);break;
        case 66 : lcd_gotoxy(0,0); lcd_putsf("Zdhin");suara(167);delay_ms(1000);
break;
        case 75 : lcd_gotoxy(0,0); lcd_putsf("Thin");suara(166); delay_ms(1000);
break;
        case 76 : lcd_gotoxy(0,0); lcd_putsf("Dhin");suara(165);delay_ms(1000);
break;
        case 82 : lcd_gotoxy(0,0); lcd_putsf("Shin");suara(164);delay_ms(1000);
break;
        case 93 : lcd_gotoxy(0,0); lcd_putsf("Syin");suara(163); delay_ms(1000);
break;

```

```

        case 33 : lcd_gotoxy(0,0); lcd_putsf("Yin");suara(180); delay_ms(1000);
break;    //deret kedua
        case 42 : lcd_gotoxy(0,0); lcd_putsf("in");suara(179); delay_ms(1000);break;
        case 50 : lcd_gotoxy(0,0); lcd_putsf("lin");suara(178);delay_ms(1000); break;
        case 49 : lcd_gotoxy(0,0); lcd_putsf("Hin");suara(177); delay_ms(1000);
break;
        case 58 : lcd_gotoxy(0,0); lcd_putsf("Win");suara(176);delay_ms(1000);
break;
        case 65 : lcd_gotoxy(0,0); lcd_putsf("Nin");suara(175);delay_ms(1000);
break;

    }
}
}
void dhomah_tain()
{
    {switch (a)
    {

        case 21 : lcd_gotoxy(0,0); lcd_putsf("Sun");suara(192); delay_ms(1000);
break;    //deret pertama
        case 29 : lcd_gotoxy(0,0); lcd_putsf("Zun");suara(191); delay_ms(1000);
break;
        case 36 : lcd_gotoxy(0,0); lcd_putsf("Run");suara(190); delay_ms(1000);
break;
        case 45 : lcd_gotoxy(0,0); lcd_putsf("Dzun");suara(189); delay_ms(1000);
break;
        case 44 : lcd_gotoxy(0,0); lcd_putsf("Dun");suara(188);delay_ms(1000);
break;
        case 53 : lcd_gotoxy(0,0); lcd_putsf("Khun");suara(187);delay_ms(1000);
break;
        case 60 : lcd_gotoxy(0,0); lcd_putsf("Khun");suara(186);delay_ms(1000);
break;
        case 67 : lcd_gotoxy(0,0); lcd_putsf("Jun");suara(185); delay_ms(1000);
break;
        case 68 : lcd_gotoxy(0,0); lcd_putsf("Tsun");suara(184);delay_ms(1000);
break;
        case 77 : lcd_gotoxy(0,0); lcd_putsf("Tun");suara(183); delay_ms(1000);
break;
        case 84 : lcd_gotoxy(0,0); lcd_putsf("Bun");suara(182);delay_ms(1000);
break;
        case 91 : lcd_gotoxy(0,0); lcd_putsf("Un");suara(181);delay_ms(1000); break;

        case 28 : lcd_gotoxy(0,0); lcd_putsf("Mun");suara(204); delay_ms(1000);
break;    //deret kedua
        case 27 : lcd_gotoxy(0,0); lcd_putsf("Lun");suara(203); delay_ms(1000);
break;

```

```

        case 35 : lcd_gotoxy(0,0); lcd_putsf("Kun");suara(202); delay_ms(1000);
break;
        case 43 : lcd_gotoxy(0,0); lcd_putsf("Qun");suara(201); delay_ms(1000);
break;
        case 52 : lcd_gotoxy(0,0); lcd_putsf("Fun");suara(200);delay_ms(1000); break;
        case 51 : lcd_gotoxy(0,0); lcd_putsf("Ghun");suara(199); delay_ms(1000);
break;
        case 59 : lcd_gotoxy(0,0); lcd_putsf("Aun");suara(198);
delay_ms(1000);break;
        case 66 : lcd_gotoxy(0,0); lcd_putsf("Zdhun");suara(197);delay_ms(1000);
break;
        case 75 : lcd_gotoxy(0,0); lcd_putsf("Thun");suara(196);delay_ms(1000);
break;
        case 76 : lcd_gotoxy(0,0); lcd_putsf("Dhun");suara(195); delay_ms(1000);
break;
        case 82 : lcd_gotoxy(0,0); lcd_putsf("Shun");suara(194); delay_ms(1000);
break;
        case 93 : lcd_gotoxy(0,0); lcd_putsf("Syun");suara(193); delay_ms(1000);
break;

        case 33 : lcd_gotoxy(0,0); lcd_putsf("Yun");suara(210);delay_ms(1000);
break;      //deret kedua
        case 42 : lcd_gotoxy(0,0); lcd_putsf("un");suara(209);delay_ms(1000); break;
        case 50 : lcd_gotoxy(0,0); lcd_putsf("lun");suara(208);delay_ms(1000); break;
        case 49 : lcd_gotoxy(0,0); lcd_putsf("Hun");suara(207); delay_ms(1000);
break;
        case 58 : lcd_gotoxy(0,0); lcd_putsf("Wun");suara(206); delay_ms(1000);
break;
        case 65 : lcd_gotoxy(0,0); lcd_putsf("Nun");suara(205); delay_ms(1000);
break;

    }
}
}

```

// Declare your global variables here

```
void main(void)
```

```
{
```

// Declare your local variables here

// Input/Output Ports initialization

// Port A initialization

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In

Func0=In

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTA=0x00;


```

DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;
UCSRA=0x00;
UCSRB=0x08;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x4D;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Falling Edge
// INT1: Off
// INT2: Off
GICR|=0x40;
MCUCR=0x02;
MCUCSR=0x00;
GIFR=0x40;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled

```

```

SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 8
lcd_init(16);
lcd_gotoxy(0,0);
lcd_Putsf("Media Pembelajaran");
lcd_gotoxy(1,0);
lcd_putsf("Huruf Hijaiyah");
suara(1);
delay_ms(1000);
lcd_clear();
// Global enable interrupts
#asm("sei")

while (1)
{
    lcd_clear();
    a=PORTA;
    if (PINA.0==1)
    {
        fathah();

    }
    else if (PINA.1==1)
    {
        kasroh();

    }
    else if (PINA.2==1)
    {
        dhomah();

    }
}

```

```

else if (PINA.4==1)
{
    fathah_tain();
}
else if (PINA.5==1)
{
    kasroh_tain();
}
else if (PINA.6==1)
{
    dhomah_tain();
}

else if (PINA.0==0 && PINA.1==0 && PINA.2==0 && PINA.3==0 &&
PINA.4==0)
{
    hijaiyah();
}

delay_ms(300);
lcd_clear();
}

}

```

**Data Santri TKA/TPA Masjid Al-Falaah Mrican, Kelas Aisyah Belajar
Mengenal Huruf Hijaiyah dengan Menggunakan Media
Pembelajaran Huruf Hijaiyah**













Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 16 Kbytes of In-System Self-programmable Flash program memory
 - 512 Bytes EEPROM
 - 1 Kbyte Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels In TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
 - 2.7V - 5.5V for ATmega16L
 - 4.5V - 5.5V for ATmega16
- Speed Grades
 - 0 - 8 MHz for ATmega16L
 - 0 - 16 MHz for ATmega16
- Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L
 - Active: 1.1 mA
 - Idle Mode: 0.35 mA
 - Power-down Mode: < 1 µA



8-bit AVR®
Microcontroller
with 16K Bytes
In-System
Programmable
Flash

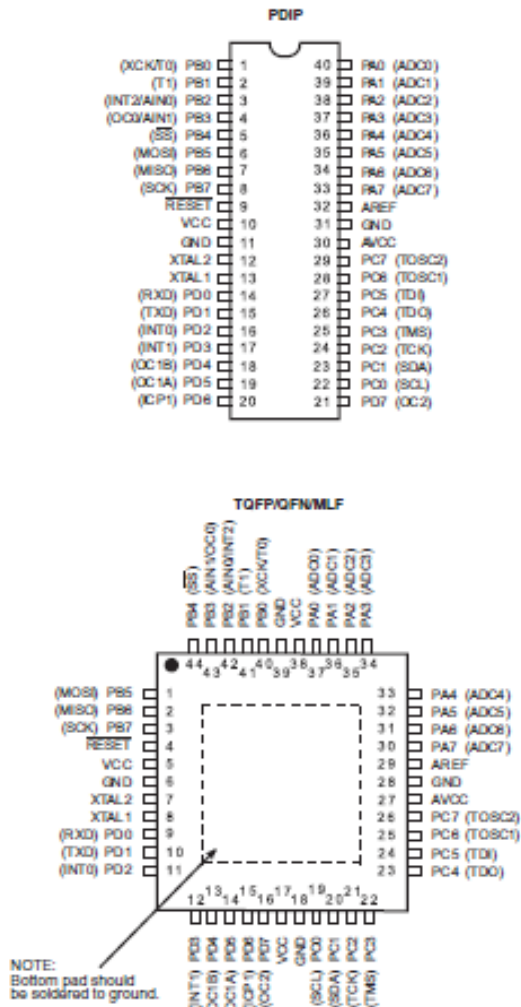
ATmega16
ATmega16L

Rev. 2486T-AVR-07/10



Pin Configurations

Figure 1. Pinout ATmega16



Disclaimer

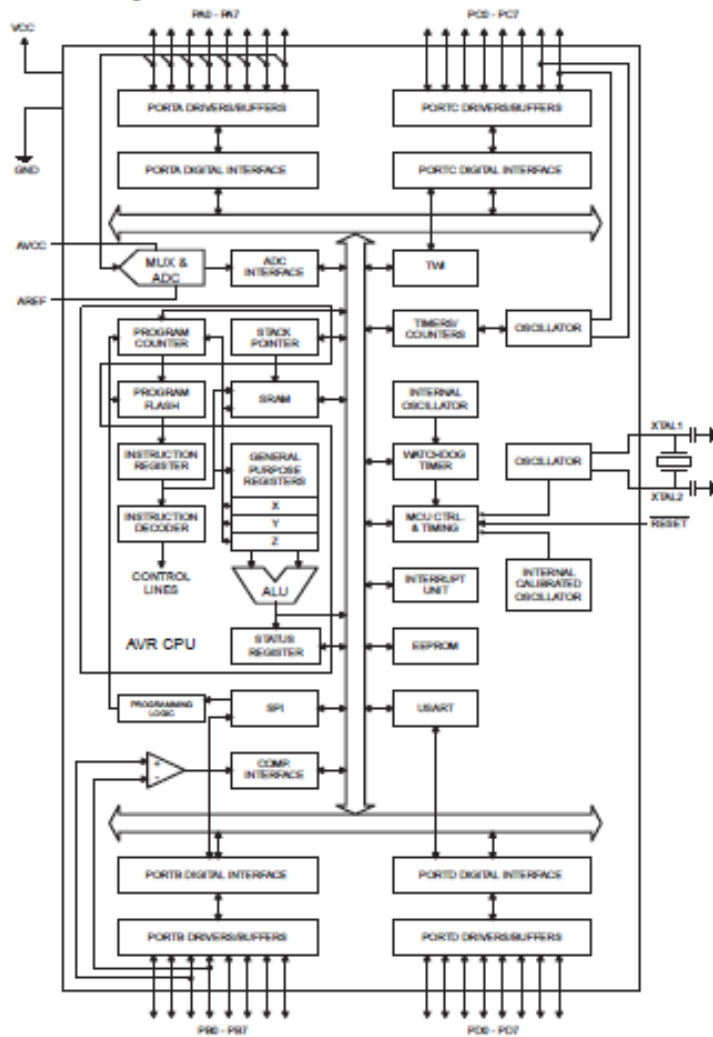
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Overview

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega16 provides the following features: 16 Kbytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1 Kbyte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega16 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Pin Descriptions

VCC Digital supply voltage.

GND Ground.

Port A (PA7..PA0) Port A serves as the analog inputs to the A/D Converter.

Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B (PB7..PB0)	<p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port B also serves the functions of various special features of the ATmega16 as listed on page 58.</p>
Port C (PC7..PC0)	<p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.</p> <p>Port C also serves the functions of the JTAG interface and other special features of the ATmega16 as listed on page 61.</p>
Port D (PD7..PD0)	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega16 as listed on page 63.</p>
RESET	<p>Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 38. Shorter pulses are not guaranteed to generate a reset.</p>
XTAL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the inverting Oscillator amplifier.
AVCC	AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.
AREF	AREF is the analog reference pin for the A/D Converter.



Resources

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.

Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C Compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C Compiler documentation for more details.



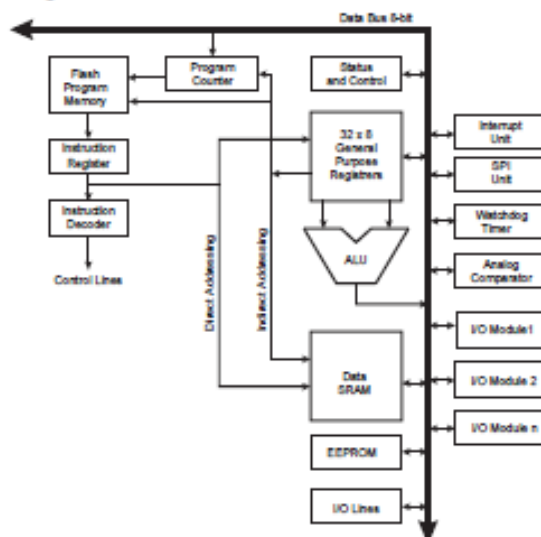
AVR CPU Core

Introduction

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Architectural Overview

Figure 3. Block Diagram of the AVR MCU Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of these address pointers can also be used as an address pointer for look up tables in Flash Program memory. These added function registers are the 16-bit X-register, Y-register, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16-bit or 32-bit instruction.

Program Flash memory space is divided in two sections, the Boot program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The Stack Pointer SP is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the Status Register. All interrupts have a separate interrupt vector in the interrupt vector table. The interrupts have priority in accordance with their interrupt vector position. The lower the interrupt vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, \$20 - \$5F.

ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the "Instruction Set" section for a detailed description.

Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7 – I: Global Interrupt Enable

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The Bit Copy instructions BLD (Bit Load) and BST (Bit Store) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry is useful in BCD arithmetic. See the "Instruction Set Description" for detailed information.

- **Bit 4 – S: Sign Bit, $S = N \oplus V$**

The S-bit is always an exclusive or between the Negative Flag N and the Two's Complement Overflow Flag V. See the "Instruction Set Description" for detailed information.

- **Bit 3 – V: Two's Complement Overflow Flag**

The Two's Complement Overflow Flag V supports two's complement arithmetics. See the "Instruction Set Description" for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

General Purpose Register File

The Register File is optimized for the AVR Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Figure 4 shows the structure of the 32 general purpose working registers in the CPU.

Figure 4. AVR CPU General Purpose Working Registers

	7	0	Addr.	
General Purpose Working Registers	R0		\$00	
	R1		\$01	
	R2		\$02	
	...			
	R13		\$0D	
	R14		\$0E	
	R15		\$0F	
	R16		\$10	
	R17		\$11	
	...			
	R26		\$1A	X-register Low Byte
	R27		\$1B	X-register High Byte
	R28		\$1C	Y-register Low Byte
	R29		\$1D	Y-register High Byte
	R30		\$1E	Z-register Low Byte
	R31		\$1F	Z-register High Byte

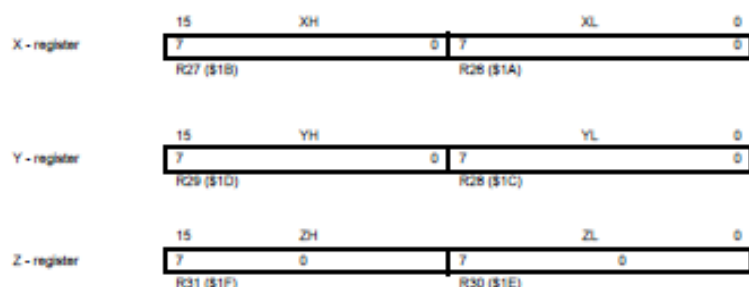
Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 4, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y-, and Z-pointer Registers can be set to index any register in the file.

The X-register, Y-register and Z-register

The registers R26..R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the Data Space. The three indirect address registers X, Y, and Z are defined as described in Figure 5.

Figure 5. The X-register, Y-register, and Z-register



In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the Instruction Set Reference for details).

Stack Pointer

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. The Stack Pointer Register always points to the top of the Stack. Note that the Stack is implemented as growing from higher memory locations to lower memory locations. This implies that a Stack PUSH command decreases the Stack Pointer. If software reads the Program Counter from the Stack after a call or an interrupt, unused bits (15:13) should be masked out.

The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above \$60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when the return address is pushed onto the Stack with subroutine call or interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when data is popped from the Stack with return from subroutine RET or return from interrupt RETI.

The AVR Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Instruction Execution Timing

This section describes the general access timing concepts for instruction execution. The AVR CPU is driven by the CPU clock clk_{CPU} , directly generated from the selected clock source for the chip. No internal clock division is used.

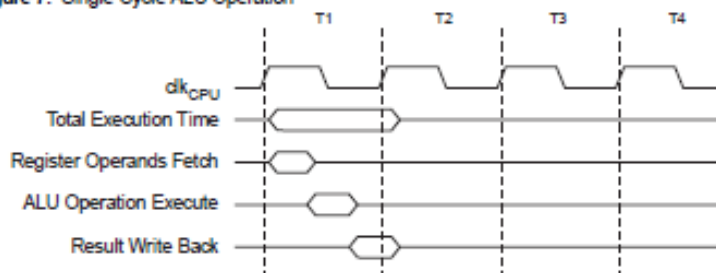
Figure 6 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

Figure 6. The Parallel Instruction Fetches and Instruction Executions



Figure 7 shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 7. Single Cycle ALU Operation



Reset and Interrupt Handling

The AVR provides several different interrupt sources. These interrupts and the separate reset vector each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt. Depending on the Program Counter value, interrupts may be automatically disabled when Boot Lock bits BLB02 or BLB12 are programmed. This feature improves software security. See the section "Memory Programming" on page 259 for details.

The lowest addresses in the program memory space are by default defined as the Reset and Interrupt Vectors. The complete list of vectors is shown in "Interrupts" on page 45. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INTO – the External Interrupt Request



WT5001M02-28P

语音模块展示

5-30 minutes of the most competitive high-quality MP3 module

Introduction

WT5001M02-28P is a support TF card and SPI-FLASH as a storage medium MP3 player module, supports wav and mp3 files, professional sound quality, comes with one watt amplifier, can be downloaded online and offline with the TF card to copy files; support RS232 serial port control, support U disk from the extended to playback, has a unique function of the music into the player. Can be flexibly applied to explain the kind of variety of voice applications.

▲ custom development for various functions.



Features

- Can be accessed 1W5Q speaker (comes with a 1W power amplifier);
- With BUSY output;
- Support for wav, mp3 format;
- Support for SPI-FLASH download;
- One ADC button interface, support for five standard MP3 features buttons;
- With a special function keys, each time to achieve play the next, and single loop the current track, Long press and TF card can copy the contents of the FLASH;

- Support for RS232 serial communication;
- With a TF card socket, support TF card player;
- With USB interface pins, support U disk playback;
- Offline support TF card or U disk copy the contents to the SPI-FLASH, copy the file can be changed by setting iSound.mp3 power play mode;
- Long time stability, anti-interference ability;
- ▲ Custom development for various functions.

Model	Package	Capacity	Operating Voltage	Size	Operating Temp
WT5001M02-28P	DIP28	SPI-FLASH 4MB~64MB TF card : 128MB~2GB	3.3V-5.2V	35.1mm*19.7mm	-30~+75℃

Application Diagram



Application Diagram



WT5001M02-28P Minimum application schematic

Application

WT5001M01-16P module can be applied in the automotive, intelligent home systems, home burglar alarm, voice prompt medical devices, personalized audio players, home appliances, entertainment, learning model, intelligent transportation equipment, industrial control field, toys and other fields. For voice needs to be replaced or longer, higher quality requirements of the occasion.

How to Use an MMC

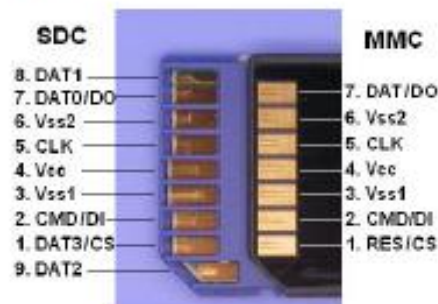
Apr 17, 2006



Now *SD Memory Card* (Secure Digital Memory Card) is the most popular memory card for mobile equipments. The SD Memory Card (SDC below) was developed as upper-compatible to *Multi Media Card* (MMC below) so that the SDC compliant equipments can also use an MMC with a few considerations. There are also reduced size versions, such as *RS-MMC*, *miniSD* and *microSD*, with same function. The MMC/SDC has a microcontroller in it, the flash memory controls (erase, read, write and error control) are completed at inside of the memory card. The data is transferred between memory card and host controller in unit of 512 bytes per block in default, so that it can be seen like a generic hard disk drive from view point of application programs. The currently defined file system is only FAT12/16 with FDISK partitioning rule. The FAT32 is defined for only high capacity ($\geq 4G$) cards.

This page describes the basic knowledge and miscellaneous things that I become aware, on using MMC/SDC with small embedded system. I believe that this information will be a useful getting started notes for people who is going to begin to enjoy MM/SDC.

Contact Surface



Right photo shows the contact surface of the SDC/MMC. The MMC has seven contact pads and the SDC has nine contact pads that two pads added to MMC. Three of the contacts for each occupy as power supply pins so that the effective signal numbers are four and six. Ofcourse the data transfer between the host and the card is done in clocked serial data transfer.

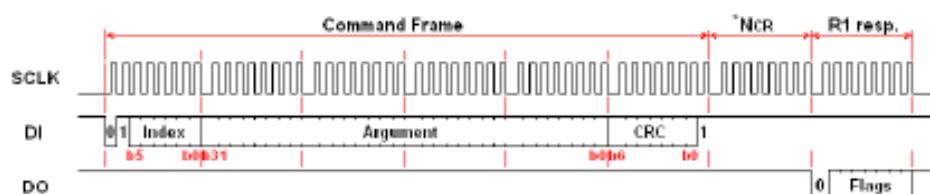
The working supply voltage range is indicated in OCR register and it should be read to confirm the operating voltage range. However, the supply voltage can be fixed to a proper value because the MMC/SDC works at supply voltage of 2.7 to 3.6 volts at least. The current consumption can reach up to several ten milliamperes at worst case, the host system should be able to supply 100 milliamperes.

SPI Mode

SPI mode is an alternative operating mode that defined to use MMC/SDC without its specific host interface. The communication protocol for the SPI mode is very simple compared to MMC/SDC native mode, the MMC/SDC can be attached via a generic SPI port or a GPIO port built in most microcontrollers. Therefore the SPI mode is suitable for low cost embedded applications. Especially, *there is no reason to use native mode for electronic handiwork as a hobby*. For SDC, the 'SPI mode 0' is defined for its SPI mode. But for MMC, it is not the SPI timing, both latch and shift actions are defined with rising edge of SCLK, but it seems work in SPI mode 0 at SPI mode. Thus *SPI Mode 0* (positive clock, front edge latch, back edge shift) is the proper setting for MMC/SDC interface, and SPI mode 3 also works as well.

Command and Response

In SPI mode, the data direction on the signal line is fixed and the data is transferred in byte oriented serial communication. The command frame from host to card is a fixed length (six bytes) packet that shown below. When a command frame is transmitted to the card, a response to the command (R1, R2 or R3) will be sent back to the host. Because data transfer is driven by serial clock generated by host, the host must continue to read bytes until receive an effective response. The command response time (NCR) is 0 to 8 bytes for SDC, 1 to 8 bytes for MMC. The CS signal must be held low during a transaction (command, response and data transfer if exist). The CRC field is optional in SPI mode, but it is required as a bit field to compose a command frame.



SPI Command Set

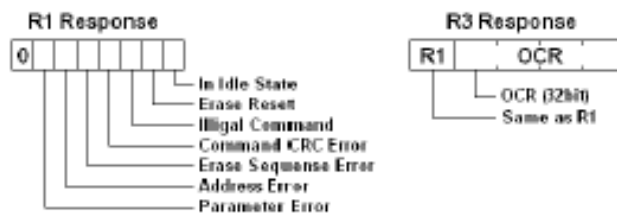
Each command is expressed in abbreviation like GO_IDLE_STATE or CMD<nr>, <nr> is the number of the command index and the value can be 0 to 63. Following table describes only commands that to be usually used for generic read/write and card initialization. For details on all commands, please refer to spec sheets from MMCA and SDCA.

Command Index	Argument	Response	Data	Abbreviation	Description
CMD0	None(0)	R1	No	GO_IDLE_STATE	Software reset.
CMD1	None(0)	R1	No	SEND_OP_COND	Initiate initialization process.
ACMD41(*1)	None(0)	R1	No	APP_SEND_OP_COND	For only SDC. Initiate initialization process.
CMD9	None(0)	R1	Yes	SEND_CSD	Read CSD register.
CMD10	None(0)	R1	Yes	SEND_CID	Read CID register.
CMD12	None(0)	R1b	No	STOP_TRANSMISSION	Stop to read data.
CMD16	Block length[31:0]	R1	No	SET_BLOCKLEN	Change R/W block size.

CMD17	Address[31:0]	R1	Yes	READ_SINGLE_BLOCK	Read a block.
CMD18	Address[31:0]	R1	Yes	READ_MULTIPLE_BLOCK	Read multiple blocks.
CMD23	Number of blocks[15:0]	R1	No	SET_BLOCK_COUNT	For only MMC. Define number of blocks to transfer with next multi-block read/write command.
ACMD23(*1)	Number of blocks[22:0]	R1	No	SET_WR_BLOCK_ERASE_COUNT	For only SDC. Define number of blocks to pre-erase with next multi-block write command.
CMD24	Address[31:0]	R1	Yes	WRITE_BLOCK	Write a block.
CMD25	Address[31:0]	R1	Yes	WRITE_MULTIPLE_BLOCK	Write multiple blocks.
CMD55(*1)	None(0)	R1	No	APP_CMD	Application specific command.
CMD58	None(0)	R3	No	READ_OCR	Read OCR.

*1:ACMD<n> means a command sequence of CMD55-CMD<n>.

SPI Response



There are three command response formats, *R1*, *R2* and *R3*, depends on each command. A byte of response *R1* is returned for most commands. The bit field for *R1* response is shown in right image, and a 0x00 zero means successful. When any error occurred, corresponding bit will be set. The *R3* response is for only CMD58, its first byte is same as *R1* and is trailing the content of OCR.

Some command takes a time longer than NCR and it responds *R1b* that is *R1* response followed by busy flag (*DO* is held low as long as internal process is being executed). The host controller should wait for end of the process until 0xFF is received.

Initialization Procedure for SPI Mode

After power on reset, MMC/SDC enters its native operating mode. To put it SPI mode, following procedure must be performed.

Power ON (Insertion)

After supply voltage reached 2.2 volts, wait for a millisecond at least, set *DI* and *CS* high and apply more than 74 pulses to *SCLK* and the card will able to accept a native command.

Software Reset

Send a *CMD0* with *CS* low to reset the card. The card samples *CS* signal when a *CMD0* is detected. If the *CS* signal is low, the card enters SPI mode. Since the *CMD0* must be sent as a native command, the CRC field must have a valid value. When once the card enters SPI mode, the CRC check is disabled and the CRC value is negligible, so that

command transmission routine can be written with the hardcoded CRC byte of 0x95 that valid for only CMD0. When the CMD0 is accepted successfully, the card will enter idle state and respond R1 response with In Idle State bit (0x01). The CRC can also be re-enabled with CMD59.

Initialization

In idle state, the card accept only CMD0, CMD1 and CMD58. Any other commands will be rejected. When the card detects a *CMD1*, it starts initialization. To poll end of the initialization, host controller must repeat to send CMD1 and check the response. When the card is initialized successfully, In Idle State in the R1 response is cleared (0x00). The initialization process can take *several hundred milliseconds* (large cards tend to longer), so that this is a consideration to determine the time out value. After the initialization has finished, generic read/write will be accepted. In this time, OCR and CID may be read to confirm operating voltage range, card size or any properties if needed.

For SDCs, *ACMD41* instead of CMD1 is recommended to initiate initialization. The CMD1 seems to not work on some SDCs, so that a retry sequence with ACMD41 should be done when CMD1 is rejected or a time out is occurred during idle polling.

Data Transfer

Data Packet and Data Response

Data Packet

Data Token	Data Block	CRC
1 byte	1 ~ 2048 bytes	2 bytes

Data Token

1 1 1 1 1 1 1 0	Data token for CMD17/18/24
1 1 1 1 1 1 0 0	Data token for CMD25
1 1 1 1 1 0 1 1	Stop Tran token for CMD25

Data Response

X	X	X	0	Status	1
0	1	0	—	Data accepted	
1	0	1	—	Data rejected due to a CRC error	
1	1	0	—	Data rejected due to a write error	

Error Token

0	0	0	Flags
			Error
			CC error
			Card ECC failed
			Out of range
			Card is locked

In a transaction with data transfer, one or more data blocks will be sent/received after command response. The data block is transferred as a data packet that consist of Token, Data Block and CRC. The format of the data packet is shown in right image and there are three data tokens. As for Stop Tran token that means end of multiple block write, it is used in single byte without data block and CRC.

Single Block Read



The argument specifies the location to start to read *in unit of byte*. The sector address specified by upper layer must be